PARTISN USER'S GUIDE

Transport Methods Group, CCS-4
Los Alamos National Laboratory

2

CCS-4 — Transport Methods Group

Los Alamos
National Laboratory

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

An Affirmative Action/Equal Opportunity Employer

This work was supported by the US Department of Energy.

USER'S GUIDE FOR PARTISN: A CODE PACKAGE FOR PARALLEL, TIME-DEPENDENT SN TRANSPORT

by
Ray E. Alcouffe, Randal S. Baker, Jon A. Dahl,
and Scott A. Turner

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The PARTISN code package is a modular computer program designed to solve the multi-dimensional, time-independent or dependent, multigroup discrete-ordinates form of the Boltzmann transport equation.[1,2]

PARTISN™* is based on the modular construction of the DANTSYS™[,3] code system package. This modular construction separates the input processing, the transport equation solving, and the postprocessing, or edit functions, into distinct, independently executable code modules, the INPUT, SOLVER, and EDIT modules, respectively. These modules are connected to one another solely by means of binary interface files. Standardized data and file management techniques as defined[4] and developed by the Committee Computer Code Coordination (CCCC) are used for these files.

Some of the major features included in the PARTISN package are:

1. a free-field format text input capability, designed with the user in mind,
2. the use of diffusion synthetic acceleration[5] or transport synthetic acceleration[6] to accelerate the iterative process in the SOLVER module,
3. direct (forward) or adjoint calculational capability,
4. 1-D (slab, two-angle slab, cylindrical, and spherical), 2D (x-y, r-z, and r-theta) and 3-D (x-y-z and r-z-theta) geometry options,
5. arbitrary anisotropic scattering order via standard PN expansions or Galerkin scattering,
6. vacuum, reflective, periodic, white, or surface source boundary condition options,
7. inhomogeneous (fixed) source or k_{eff} calculation options, as well as time-absorption (α), nuclide concentration, or dimensional search options,
8. "diamond-differencing," adaptive weighted diamond differencing (AWDD), linear discontinuous or exponential discontinuous spacial differencing schemes for the solution of the transport equation,
9. a diffusion solver that uses the multigrid[7] or conjugate gradient method,

* - DANTSYS and PARTISN are trademarks of the Regents of the University of California, Los Alamos National Laboratory

10. user flexibility in using either ASCII text or sequential file input,
11. a ray trace first collision option to obtain a first collision source from an arbitrary source distribution (volume and external boundary sources),[8]
12. user flexibility in controlling the execution of both modules and submodules,
13. extensive, user-oriented error diagnostics,
14. a strictly positive scattering source[9] option,
15. an automatic mesh coarsening[10] option,
16. block Adaptive Mesh Refinement (AMR),[11]
17. time-dependent calculations,[12] and
18. parallel processing.[13-15]

DOCUMENTATION FOR PARTISN USAGE

The documentation described here constitutes a complete manual for the use of the PARTISN code.

What Is In This User's Guide

This User's Guide is a chapter from the much larger PARTISN system document. This Guide provides the ASCII text input specifications for PARTISN.

The guide is intended to serve as a complete input manual for two classes of user. Special, succinct sections containing summaries and compact tables are intended for the advanced user in order to make his input preparation more efficient. The main body of the guide concerns itself with descriptions of the input and should be sufficient for the user familiar with discrete ordinates concepts. Novice users may find other chapters of the document necessary.

This Guide first gives an overview of the input block order required by the code.

Next is a "mini-manual" in which are listed all the names of available input arrays arranged by input block. Definitions of input arrays are not given, as the names are suggestive, but expected types and sizes are provided. This mini-manual is very useful to the user as a quick check for completeness, a quick reference to type and size, and as an index into the more detailed array descriptions that follow. For the experienced user, the mini-manual is frequently all that is needed to prepare a complete input deck.

Following the mini-manual are reference sections describing in detail all the input parameters and arrays.

Appendix A provides a sample PARTISN problem with explanation of the output for the user.

Lastly, Appendix B details operating system specifics, including how to effect an execution of the code.

Information of a reference, background, or theoretical nature that the first time user may need may not be found in this User's Guide, but the user will encounter liberal references to other chapters of this document for that sort of information.

What Is Available Elsewhere

In addition to this User's Guide, the user, especially the first time user, may find the information below described in other chapters of this document pertinent. For even greater detail on some of the general items, particularly the methods items, the user should look at Ref. 16.

The chapter "DETAILS OF THE BLOCK-I, GEOMETRY, AND SOLVER INPUT" starting on page 3-1 discusses in more detail the geometry and solver concepts and their related input. If the User's Guide proves insufficient for your needs, look in this chapter. Among the many sections of the chapter are ones on the input of inhomogeneous sources and a discussion of eigenvalue searches. There is also more detail on the Block-I input.

A discussion of how the EDIT module works and more detail on preparing the input is given in the chapter "RUNNING THE EDIT MODULE" starting on page 4-1.

The chapter "FREE FIELD INPUT REFERENCE" starting on page 5-1 serves as the reference manual for the free-field input (rules, format, and operators) used in this code. That chapter is summarized in this guide, but should the summary prove inadequate, the user is referred there for full details.

The chapter "CROSS-SECTION LIBRARIES" starting on page 6-1 gives details of the many library formats available to PARTISN, including sections on how to prepare your own card-image (or text) libraries.

The chapter "MATERIAL MIXING TUTORIAL" starting on page 7-1 describes the mixing concepts in detail and shows some examples.

Next is the chapter "PARTISN — METHODS MANUAL" starting on page 8-1. That chapter describes the theoretical basis for the PARTISN code.

In the chapter "PARTISN —CODE STRUCTURE" starting on page 9-1 is shown a brief overview of the code package. Included are sections on programming practices and standards, code package structure, and functional descriptions of the three principal modules comprising the package. In particular, the code package structure must be understood in order to make up input for piecewise executions of the code that are possible with controls that are part of the input in Block-I.

Error diagnostics that the user might encounter are found in the chapter "ERROR MESSAGES" starting on page 10-1. Several examples of input errors and the resulting error messages are provided for the user.

The chapter "FILE DESCRIPTIONS" starting on page 11-1 is a reference that describes all the files used by the package. Included is a detailed description of the file structure of the code dependent, binary, sequential interface files generated by and used in the PARTISN code package. Also included are descriptions of any other files produced or used by the package, both binary and text. In some cases, this may simply be a reference to a more comprehensive document, such as the file descriptions for the CCCC standard interface files.

PARTISN INPUT OVERVIEW

Input Block Order

The full PARTISN input consists of a title line section, followed by six blocks of free field input. The title line section is not free field. Any input referred to as a block uses the free field input form.

Block-I consists of basic control and dimensional information that allows efficient packing of the array data. This information also allows checking of the lengths of arrays supplied by interface files.

Block-II contains the geometric information.

Block-III consists of the nuclear data specifications.

Block-IV contains mixing information.

Block-V contains the rest of the input needed for specifying the flux calculation.

And lastly, Block-VI contains the edit (i.e., report writing) specifications.

If a text cross-section library is to be included in the input deck, it should be placed between Blocks III and IV. PARTISN supports many library formats and so the library may or may not be in free field format depending upon the option chosen.

A full input would then look like that diagrammed on the following page.

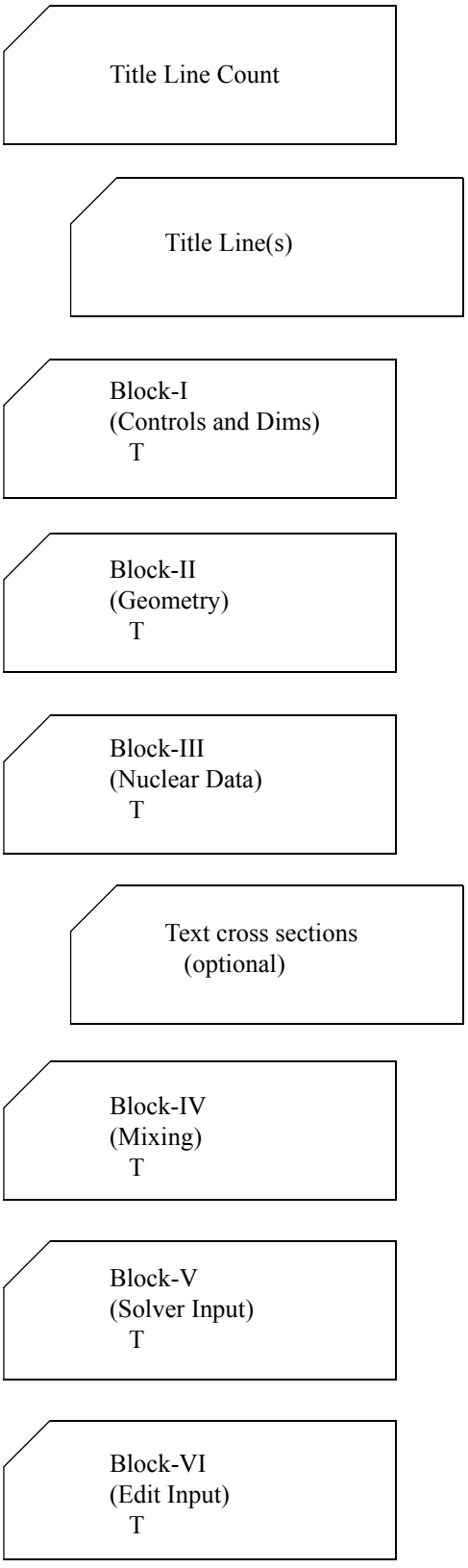


Figure 2.1 PARTISN Input Order

Free Field Input Summary

The chapter "FREE FIELD INPUT REFERENCE" starting on page 5-1 is summarized here for quick reference.

There are four basic input quantities in the free field input used in PARTISN; they are ARRAY, DATA ITEM, BLOCK, and STRING. Each of these is briefly described below along with the concept of an input operator.

Arrays

The "Array" is the most basic concept in the input. Data are given to the code by placing data items in an "Array." To make an input to an array, one simply spells out the array name, appends an equal sign, and follows that with the data items to be entered into the array. For example, input for the x distribution of the volumetric source, for which the unique array name is SOURCX, might look like:

```
SOURCX=  0 0  0  1.1 1.1  0 0 0 0 0
```

The above input would enter source values of zero for the first three intervals, 1.1 for the next 2 intervals, and then fill the rest of the ten positions in the array with zero.

Data items within an array are separated by blanks or commas. In general, blanks may be used freely throughout except within a data item, within an array name, or between an array name and its equal sign.

Single value input variables are treated as arrays of unit length.

Numeric Data Items

Numeric data items follow a Fortran input convention. For example, all of the following are valid entries for the number ten:

```
10,  1.0+1,  1E1, 10.0
```

If a decimal point is not entered, it is assumed to be after the right-most digit.

Some arrays expect integer values for input. For such arrays, any input values containing a decimal point will be truncated.

Character Data Items

Character data items follow a Fortran variable name convention in that they are composed of up to eight characters, the first of which must be alphabetic with the rest alphanumeric. However, special characters and blanks may be included if the data item is surrounded by double quotes. Operators may NOT be used with character data items.

Blocks

Arrays are entered in groups called blocks. A block consists of one or more arrays (in any order) followed by the single character T. Thus T is the block delimiter.

Strings

Arrays may need to be entered in smaller pieces called strings. Strings are delimited with a semicolon(;). When there is matrix or other 2-d input, strings are frequently used to input information by row rather than for the whole 2-d array at once. The code dictates this, the user has no choice. The user is made aware of which arrays require string input through use of a certain notation, described later, in the input array descriptions.

Comments

A slash (/) may be used to enter comments in the input stream. After a slash is read no further processing of that card-image is done.

Operators

Several data operators are available to simplify the input.

The data operators are specified in the general form

n *O* d

where:

n is the "data numerator," either an integer or a blank;
O is any one of the "data operator" characters shown below; and
d is a "data entry" (may be blank for some operators).

Note: The "data operator" character must be appended to the "data numerator."

Using operators, the SOURCX input described above could more succinctly be given as:

SOURCX= 0 0 0 2R 1.1 F0

Note that the operators for FIDO-like repeat and fill were used and were appended directly to the data numerator. In general, all the FIDO[17] operators may be used in numeric entry.

A table of the most used operators is given next including brief descriptions. For full descriptions of these and a complete list of all the available operators, including the more esoteric ones, the user is referred to “FREE FIELD INPUT DETAILS” on page 5-13.

Frequently Used Operators

Operator[a]	Functionality
nR d	REPEAT the data item d, n times.
nI d	INTERPOLATE (linear) n data items between data item d and the next data item.
nC d	SCALE (multiply) the n previous entries by d.
F d	FILL the rest of the data string with the data item d.
nY m	STRING REPEAT. Repeat the previous m strings, n times.
nL d	INTERPOLATE LOGARITHMICALLY n data items between d and the next d.
nZ	ZERO. Enter the value zero n successive times.
nS	SKIP. Skip the next n data items.
nQ m	SEQUENCE REPEAT. Enter the last m entries, n more times.
nG m	SEQUENCE REPEAT WITH SIGN CHANGE. Same as the Q option but the sign of the m entries is changed every repeat.
nN m	SEQUENCE REPEAT INVERT. Same as the Q option but the order of the m entries is inverted each repeat.
nM m	SEQUENCE REPEAT INVERT WITH SIGN CHANGE. Same as N option but the sign is also changed every repeat.
nX	COUNT CHECK. Causes code to check the number of entries in the current string so far, against the number n.

a. The operator character must always be appended directly to n. d or m need not be immediately adjacent to the operator character.

MINI-MANUAL Introduction

On the following few pages is given a complete list of the input names, expected array sizes, and order within the array. No description of the array contents is given in this MINI-MANUAL as full details are given in later sections. The MINI-MANUAL is intended to serve as a quick reference for the knowledgeable user.

In both the MINI-MANUAL and in the detailed sections which follow, a shorthand form is used to indicate the size and order of the array that the code expects. This information is enclosed in square brackets immediately after the array name. Essential features are:

1. A single entry in the brackets is the array length.

2. No brackets at all indicates a simple variable (i.e., an array of unit length).

3. A dash (-) in the brackets indicates an arbitrary length.

4. A semicolon (;) indicates that the input for that array is expected in strings. To the left of the semicolon is the string length. To the right of the semicolon is the number of strings in the array.

5. If the number of strings is shown as a product, the order is important. The leftmost quantity must be exhausted first, then, the next one to the right is varied. For example, the array name for the full spatial source distribution is shown as:

   SOURCF [IT;JT*KT*NMQ]

   where - IT is the number of fine meshes in the X-direction, JT is the number of fine meshes in the Y-direction, KT is the number of fine meshes in the Z-direction, and NMQ is the number of input source moments. For this array, the first string is composed of the P_0 source values for each x mesh point in the first y mesh in the first z layer. The next string is the P_0 source values in the second y mesh in the first z layer. This process is repeated for all JT y meshes of the first z layer. Then repeat for each of the remaining z layers. Then starting again with the first y mesh in the first z layer, the P_1 source values for each x mesh are given. After all P_1 values are given, the P_2 values follow. Continue until all NMQ moments are specified.

Note: Usually, values for the quantities within brackets will have already been specified in the input. Sometimes, however, a quantity is derived from the array input itself. For instance, in this particular case, NMQ is not an input quantity; rather, the code counts the number of strings and then, knowing JT, KT, and NGROUP, deduces what NMQ must have been.

MINI MANUAL

Title Line Control
(5I6 Format)
NHEAD,NOTTY,NOLIST, NPASS, RESTART

Title Line(s)

(IF NHEAD>0)

Block-I:Controls & Dimensions

IGEOM
NGROUP
ISN
NISO
MT
NZONE
IM
IT
JM
JT
KM
KT
IQUAD
NN

MAXLCM
MAXSCM

-- -- -- -- --

FMMIX

NOSOLV
NOEDIT

NOGEOD
NOMIX
NOASG
NOMACR
NOSLNP
NOEDTT
NOADJM

T

Block-II:Geometry

XMESH [IM+1]
YMESH [JM+1]
ZMESH [KM+1]

XINTS [IM]
YINTS [JM]
ZINTS [KM]

ZONES [IM;JM*KM]
LEVELS [IM;JM*KM]
T

Block-III: Cross Sections

LIB
  valid: *ODNINP*
  XSLIB
  ISOTXS
  GRUPXS
  BXSLIB
  MACRXS
  MACBCD
  XSLIBB
  (local)*MENDF*
  (local)*MENDFG*
  (local) *NDILIB*
  alternate XSLIB name
WRITMXS
  valid: *MACBCD*
  XSLIBB
  XSLIBF
  XSLIBE

LNG
BALXS
NTICHI
CHIVEC [NGROUP]
CHIMIX
LIBNAME
-- -- -- -- -- -- --
Rest of this block is needed only for text libraries.
MAXORD
IHM
IHT
IHS
IFIDO
ITITL
I2LP1
SAVBXS
KWIKRD (default:1)
NAMES [NISO]
EDNAME [IHT-3]
NTPI [NISO]
VEL [NGROUP]
EBOUND [NGROUP+1]

  T

iff LIB= ODNINP, insert ASCII text cross sections here

Block-IV: Mixing

MATLS [-;MT]
ASSIGN [-;NZONE]

PREMIX [-;-]

ASGMOD [-;-]
CMOD

MATNAM [MT]
ZONNAM [NZONE]
MATSPEC [-]
  valid: *ATFRAC*
  WTFRAC
  ATDEN

ATWT [-]

  T

Block-V: Solver

IEVT
ISCT
ITH
IBL
IBR
IBT
IBB
IBFRNT
IBBACK

EPSI
IITL
IITM
OITM
NOSIGF

Solver (continued)
---Accelerator Controls---

SRCACC
valid: *DSA*
TSA
NO
NORM

CHI [NGROUP;M]

DIFSOL
valid: *MG*
CG3L
CG1L
RBL
MG1L
MGPLNZ
TSASN
TSAEPSI
TSAITS
TSABETA

--- K-Code Convergence ---
KCALC

--- Output Controls ---
FLUXP
XSECTP
FISSRP
SOURCP
ANGP
BALP
RAFLUX
RMFLUX
ASLEFT
ASRITE
ASBOTT
ASTOP
ASFRNT
ASBACK

--- Miscellaneous ---
LSSN
LSXS

Solver (continued)
TRCOR
valid: *DIAG*
BHS
CESARO
NO
BHGT
BWTH
NORM

CHI [NGROUP;M]

DEN [IT;JT*KT]
--- or ---

DENX [IT], DENY[JT], DENZ[KT]

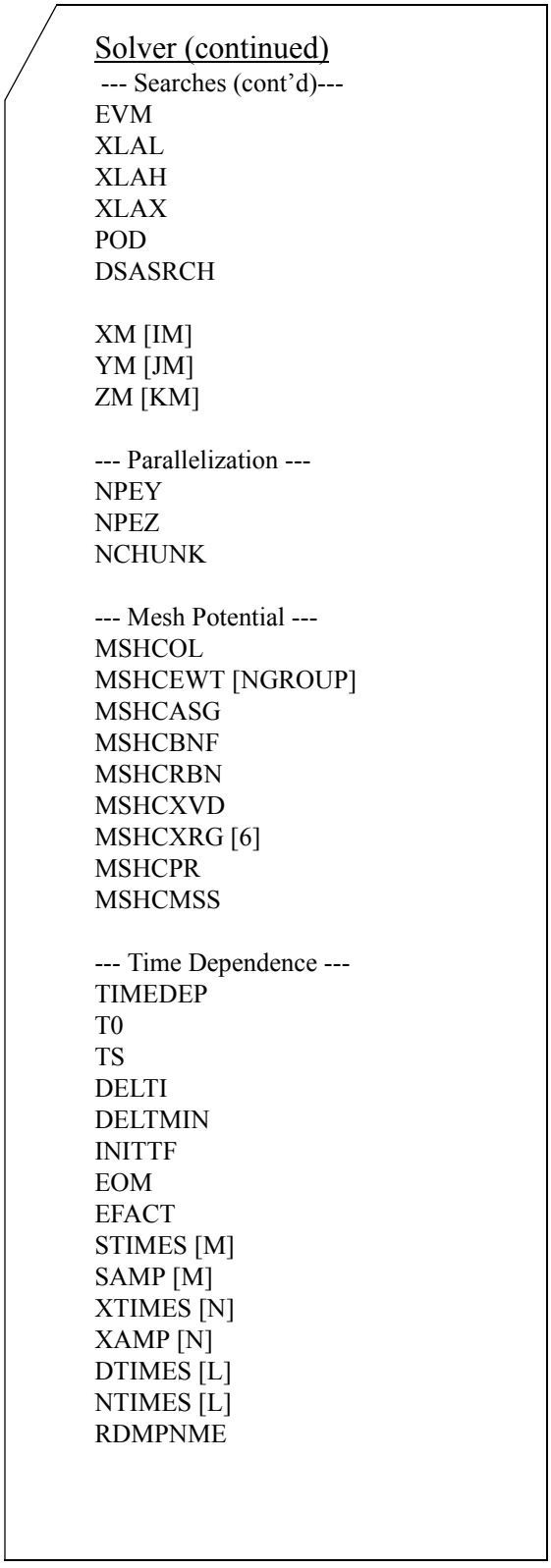
ANGFLX
FLXMOM

--- Quadrature -------
GRPSN [NGROUP]
WGTDIA
WGT [MM]
MU [MM]
ETA [MM]
NLL[NN]

--- Transport Solver ---
TRNSOL
NODAL
WDAMP [NGROUP]

--- Flux Guess -------
INFLUX

--- Searches ---------
IPVT
PV
EV


Solver (continued)
--- Searches (cont'd)---
EVM
XLAL
XLAH
XLAX
POD
DSASRCH
XM [IM]
YM [JM]
ZM [KM]
--- Parallelization ---
NPEY
NPEZ
NCHUNK
--- Mesh Potential ---
MSHCOL
MSHCEWT [NGROUP]
MSHCASG
MSHCBNF
MSHCRBN
MSHCXVD
MSHCXRG [6]
MSHCPR
MSHCMSS
--- Time Dependence ---
TIMEDEP
T0
TS
DELTI
DELTMIN
INITTF
EOM
EFACT
STIMES [M]
SAMP [M]
XTIMES [N]
XAMP [N]
DTIMES [L]
NTIMES [L]
RDMPNME

Solver (continued)

----Volumetric Source----

INSORS

SOURCE [NGROUP;NMQ]
--- or ---
SOURCX [IT;NMQ] and
SOURCY [JT;NMQ] and
SOURCZ [KT;NMQ]
--- or ---
SOURCX [IT;NMQ] and
SOURCY [JT;NMQ] and
SOURCZ [KT;NMQ] and
SOURCE [NGROUP;NMQ]
--- or ---
SOURCF [IT;JT*KT*NGROUP*NMQ]
--- or ---
SOURCF [IT;JT*KT*NMQ] and
SOURCE [NGROUP;NMQ]

-----Boundary Source-----
BSFILE [6]
Options 1: ----
SILEFT [NGROUP;JT*KT]
SIRITE [NGROUP;JT*KT]
SIBOTT [NGROUP;IT*KT]
SITOP [NGROUP;IT*KT]
SIFRNT [NGROUP;IT*JT]
SIBACK [NGROUP;IT*JT]

Options 2: ----
SALEFT [MM*4;NGROUP*JT*KT]
SARITE [MM*4;NGROUP*JT*KT]
SABOTT [MM*4;NGROUP*IT*KT]
SATOP [MM*4;NGROUP*IT*KT]
SAFRNT [MM*4;NGROUP*IT*JT]
SABACK [MM*4;NGROUP*IT*JT]

Options 3a: ----
BSLFTG [NGROUP]
BSLFTY [JT]
BSLFTZ [KT]
BSLFTA [MM*4]

BSRITG [NGROUP]
BSRITY [JT]
BSRITZ [KT]
BSRITA [MM*4]

Solver (continued)

----Boundary Source (cont'd)----

BSBOTG [NGROUP]
BSBOTX [IT]
BSBOTZ [KT]
BSBOTA [MM*4]

BSTOPG [NGROUP]
BSTOPX [IT]
BSTOPZ [KT]
BSTOPA [MM*4]

BSFRNG [NGROUP]
BSFRNX [IT]
BSFRNY [JT]
BSFRNA [MM*4]

BSBAKG [NGROUP]
BSBAKX [IT]
BSBAKY [JT]
BSBAKA [MM*4]

Options 3b: ----
BSLFTG [NGROUP]
BSLFTYZ [JT;KT]
BSLFTA [MM*4]

BSRITG [NGROUP]
BSRITYZ [JT;KT]
BSRITA [MM*4]

BSBOTG [NGROUP]
BSBOTXZ [IT;KT]
BSBOTA [MM*4]

BSTOPG [NGROUP]
BSTOPXZ [IT;KT]
BSTOPA [MM*4]

BSFRNG [NGROUP]
BSFRNXY [IT;JT]
BSFRNA [MM*4]

BSBAKG [NGROUP]
BSBAKXY [IT;JT]
BSBAKA [MM*4]

SOLVER (continued)

----Boundary Source (cont'd)----

Options 3c: ----
BSLFTG [NGROUP]
BSLFTYZ [JT;KT*MM*4]

BSRITG [NGROUP]
BSRITYZ [JT;KT*MM*4]

BSBOTG [NGROUP]
BSBOTXZ [IT;KT*MM*4]

BSTOPG [NGROUP]
BSTOPXZ [IT;KT*MM*4]

BSFRNG [NGROUP]
BSFRNXY [IT;JT*MM*4]

BSBAKG [NGROUP]
BSBAKXY [IT;JT*MM*4]

Options 3d: ----
BSLFTG [NGROUP]
SALEFT [MM*4;JT*KT]

BSRITG [NGROUP]
SARITT [MM*4;JT*KT]

BSBOTG [NGROUP]
SABOTT [MM*4;IT*KT]

BSTOPG [NGROUP]
SATOPT [MM*4;IT*KT]

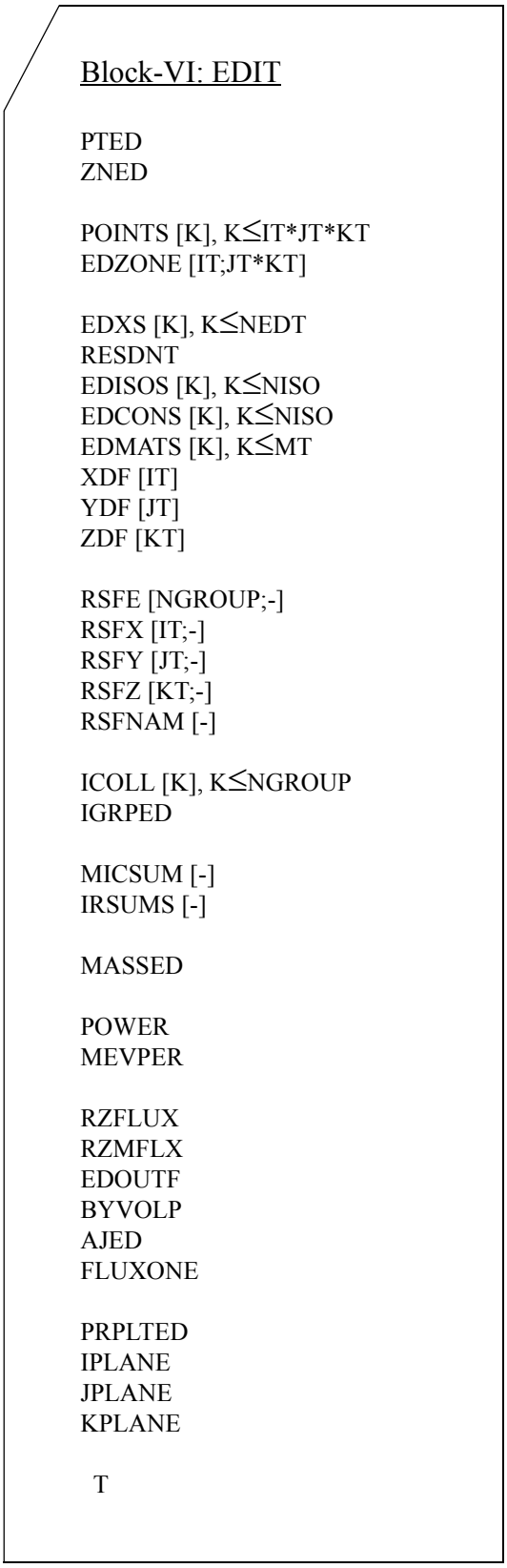
BSFRNG [NGROUP]
SAFRNT [MM*4;IT*JT]

BSBAKG [NGROUP]
SABAKT [MM*4;IT*JT]

SOLVER (continued)

--- First Collision Source ---

FCSRC
valid: *PTANA*
RAYTR
PTRAY
PTBEAML
PTBEAMR
PTBEAMB
PTBEAMT
PTBEAMF
PTBEAMK
UMCFLUX
NO
FCNRAY
FCNTR
FCWCO
FCRSTRT
FCXPOS
FCYPOS
FCZPOS
FCPOANG
FCAZANG
T


Block-VI: EDIT
PTED
ZNED
POINTS [K], K≤IT*JT*KT
EDZONE [IT;JT*KT]
EDXS [K], K≤NEDT
RESDNT
EDISOS [K], K≤NISO
EDCONS [K], K≤NISO
EDMATS [K], K≤MT
XDF [IT]
YDF [JT]
ZDF [KT]
RSFE [NGROUP;-]
RSFX [IT;-]
RSFY [JT;-]
RSFZ [KT;-]
RSFNAM [-]
ICOLL [K], K≤NGROUP
IGRPED
MICSUM [-]
IRSUMS [-]
MASSED
POWER
MEVPER
RZFLUX
RZMFLX
EDOUTF
BYVOLP
AJED
FLUXONE
PRPLTED
IPLANE
JPLANE
KPLANE
T

PARTISN INPUT DETAILS

Introduction

The following pages of this section give details for each of the input arrays. All valid PARTISN arrays are discussed in this section in detail complete enough to form the input.

However, the beginning user, particularly one unfamiliar with discrete-ordinates codes, may find that he is missing some information of a background nature. See “What Is Available Elsewhere” on page 2-14 for that.

First, here are a few general instructions:

1. All six of the input blocks are normally included. Block-I is always required but any of the other five blocks may be omitted under the proper conditions. The input module reads each block in turn and from it generates one or more binary interface files. The interface files drive the SOLVER and EDIT modules. Thus, if the user wants no edits, the Block-VI input may be omitted. Then with no interface file, the EDIT module will not be executed. Alternatively, if the interface file is available from another source, the corresponding block of input may be omitted. For instance, Block-II describes the geometry. The input module normally writes this information to the GEODST interface file. If a GEODST or a LNK3DNT file is available from another source or a previous run, the Block-II input may be omitted.

2. A general theme of the PARTISN input is that arrays that are not needed are not entered. Presence of an array indicates that it should be used. Thus, for example, if the density array is entered (DEN array), the cross section at each mesh interval will be modified accordingly. No separate switch need be set to say that the calculation should be done. To eliminate the density modification, simply remove the DEN array from the input or comment it out.

3. The arrays, in general, are grouped in the input instructions according to function. Thus, for example, the input arrays for the volumetric source are found in a single table, or grouping, of input.

4. Groupings of input data may be marked as “Required” or “Optional” in order to guide the user and speed navigation through the input instructions.

   “Required” means that at least one of the arrays in the grouping must be entered. Thus, you must read through the grouping and enter at least one of the arrays found there.

Groupings marked “Optional” may be skipped if the subject is inappropriate. Thus, using the previous example, if one has no volumetric source, one simply skips to the next grouping of input; there is no need to read about any of the arrays within the volumetric source grouping.

Arrays in groupings not marked as “Required” or “Optional” should be reviewed. These groupings contain arrays of vital data that are used in every calculation, but have default values. Thus, although you may not make any input to these arrays and they are in that sense optional, you must concern yourself with them to ensure that the default values are what is intended.

5. Input arrays may also be marked individually. If not marked, they inherit the marking of the grouping in which they are contained. Thus, an unmarked array in a “Required” grouping is required input and you must enter that array. An unmarked array in an “Optional” grouping is optional.

   You may encounter a “Required” array within an “Optional” grouping. That means that if you decide to invoke the option represented by that grouping, you must input that particular array. For example, if you want user defined response function reaction rates calculated, you must input the RSFE array.

   All arrays within unmarked groupings are optional. However, values in these arrays may be used by the code, so you should concern yourself with the default values if you choose not to enter a value.

6. Unless specifically noted otherwise, the default on all numeric inputs is zero.

7. In an adjoint run, none of the groupwise input arrays should be inverted. The code will externally identify all groups by the physical group number, not by the calculational group number (the calculational group number is in inverse order). Thus, the user interface should be consistently in the physical group order.

8. The use of information within square brackets to indicate the size of arrays and strings and the order within those arrays is the same as described in “MINI-MANUAL Introduction” on page 2-23.

9. Except where noted, arrays and strings must contain the exact number expected by the code (as indicated in the array or string description). If not, the code will eventually abort with a (hopefully) descriptive error message or messages.

10. New users reading these instructions for the first time and unfamiliar with the PARTISN input may find it helpful to follow the sample input in Appendix A while reading this section.

11. Array names are shown here in upper case. What you should actually input for them will depend upon the code’s implementation on your platform. At the present time, on most platforms, you should use lower case input.

12. Items in italics in the input instructions indicate actual values that may be entered for an array. You will frequently find switches where the input is the digit 0 or the digit 1. This will be represented by *0/1* in the input description. In other arrays where an exact character string is required such as “ISOTXS” in the LIB array, you will find the notation *ISOTXS*. Note that in this notation, the word is both upper case and italicized. This combination means you must enter exactly those characters. Again, although the characters will be shown here in upper case, what you should actually input for them will depend upon the code’s implementation on your platform.

13. When a template for the input form is given, as for the MATLS array, the style in the template tells the user what is expected. If an input word or value is lower case and italicized, the user is to replace that position with the entry of his choice. If the input word is in italicized style and in upper case, the user is to input exactly those characters to achieve the desired result. Depending on the implementation on your platform, the input word, itself, is usually in lower case.

14. Units to be used for the input quantities are not spelled out as they only need to be self consistent. However, the following are commonly used: Dimensions in centimeters, isotopic cross sections in barns per atom; then it follows that atom densities are in atoms per barn-centimeter. Sources are particles per cm^3 per second for volumetric sources and particles per cm^2 per second for boundary sources; fluxes will then be in particles per cm^2 per second.

Title Line Details

Title Line Control (format 5I6)[a] {Required}

Word	Name	Comments
1	NHEAD	Number of title lines that follow.[b]
2	NOTTY	Suppress output to on-line user terminal? *0/1* = no/yes.
3	NOLIST	Suppress listing of all ASCII text input? *0/1* = no/yes.
4	NPASS	Not used.
5	RESTART	Perform a time-dependent restart. *0/1* = no/yes.

a. WARNING! Note that this first line is in fixed format.

b. Follow this control line with NHEAD title lines containing descriptive comments. Each title line may contain up to 72 characters.

Block-I Details: Dimensions and Controls

Dimensions
{Required}

Name	Comments
IGEOM	Geometry. *1/2/3/4/6/7/11/14/15* = planar/cylindrical/spherical/two-angle slab/x-y/r-z/r-theta/x-y-z/r-z-theta or use one of the following character strings: *PLANE, CYLINDER, CYL, SPHERE, SPH, SLAB, SLAB2ANG,X-Y, R-Z, R-THETA, X-Y-Z, R-Z-THETA*
NGROUP	Number of energy groups.
ISN	S_n order to be used.
NISO	Number of physical isotopes on the basic input cross-section library.
MT	Number of physical materials[a] to be created.
NZONE	Number of geometric zones[b] in problem.
IM	Number of coarse mesh intervals[c] in the x (or r) direction.
IT	Total number of fine mesh intervals[d] in the x (or r) direction.
JM	Number of coarse mesh intervals in the y (or z) direction.
JT	Total number of fine mesh intervals in the y (or z) direction.
KM	Number of coarse mesh intervals in the z (or θ) direction.
KT	Total number of fine mesh intervals in the z (or θ) direction.

a. Material is defined on page 2-50.

b. Zone is defined on page 3-15.

c. Coarse mesh is defined on page 3-15.

d. Fine mesh is defined on page 3-15.

Block-I Details: Dimensions and Controls (Cont.)

Name	Comments
IQUAD	Source of the quadrature constants (OPTIONAL). Enter one of the following values: Value — Description *1* — Traditional built-in constants. Any even value for ISN can be used between 2 and 16 (48 for 1-D), inclusive. This is the default. *2* — For 1-D, traditional built-in DPn constants [S4, S8, S12, S16, S24, S32, S40, S48, S64, or S96]. For 2D and 3-D, same as IQUAD=1. *3* — Card input (general quadrature). ISN must be set to the number of angles/octant, NN entry is REQUIRED in Block I, and NLL entry is REQUIRED in Block V. *4* — Galerkin scattering. *5* — Triangular Chebychev-Legendre built-in set. Any even value for ISN can be used up to 50. From 50 to 100, ISN must be in multiples of 10. *6* — Rectangular Chebychev-Legendre built-in set. Any even value for ISN can be used up to 50. From 50 to 100, ISN must be in multiples of 10. *7* — Get constants from SNCONS file (triangular only). *8* — Square DPn Chebychev. *9* — 1-D Generalized Quadrature (Even moment, triangular for cylinders).
NN	Number of levels/octant. REQUIRED for IQUAD=3 only, otherwise disregarded.

Storage Requirements
{Optional}

Name	Description
MAXSCM	Length of SCM desired (default=40000_{10}).
MAXLCM	Length of LCM desired (default=140000_{10}).
	MAXSCM and MAXLCM are only used for storage in the Input and Edit modules. Solver storage is performed dynamically.

Note: The above input (Dimensions plus Storage Requirements) for Block-I will cause the code to attempt to produce a full run, subject to availability of the input normally found in the other Blocks. The controls below allow shortened print files, partial runs (say, of only the input module), or cause the code to ignore any of the other input Blocks present. For full details on their use, see "PIECEWISE EXECUTION" on page 9-21.

Fine Mesh Mixing[a]
{Optional}

Name	Description
FMMIX	Read the composition of each fine mesh from the binary file LNK3DNT. *0/1*=no/binary file LNK3DNT.

a. x-y or x-y-z geometry only.

The fine mesh mixing algorithm is designed for a general geometry option in x-y or x-y-z geometry using a volume fraction method on the fine mesh. It implies that one has an auxiliary code which will generate the volume fraction data from a general geometry description and store it on the binary file LNK3DNT.

Run Configuration Controls
{Optional}

Name	Comments
NOSOLV	Suppress solver module execution. *0/1*=no/yes.
NOEDIT	*S*uppress edit module execution. *0/1*=no/yes.
NOGEOD	Suppress writing GEODST file even though the geometry input (Block-II) may be present. *0/1*=no/yes.
NOMIX	Suppress writing mixing files even though the mixing input in Block-IV may be present. *0/1*=no/yes.
NOASG	Suppress writing ASGMAT file even though the assignment input in Block-IV may be present. *0/1*=no/yes.
NOMACR	Suppress writing the MACRXS file even though both Block-III and Block-IV may be present. *0/1*=no/yes.
NOSLNP	Suppress writing the SOLINP file even though Block-V may be present. *0/1*=no/yes.
NOEDTT	Suppress writing the EDITIT file even though Block-VI may be present. *0/1*=no/yes.
NOADJM	Suppress writing the ADJMAC file even though an adjoint calculation is called for. *0/1*=no/yes.

Note: Default on all these controls is no.

Block-II Details: Geometry

Geometry Arrays[a] {Required}

Name	Comments
XMESH [IM+1]	x (or r) coordinates of coarse mesh edges.
YMESH [JM+1]	y (or z) coordinates of coarse mesh edges.
ZMESH [KM+1]	z (or θ) coordinates of coarse mesh edges. When r-z-θ geometry used, θ is in revolutions. For r-θ geometry, use YMESH to enter the θ coordinates.
XINTS [IM]	Number of fine meshes in each coarse x (r) mesh
YINTS [JM]	Number of fine meshes in each coarse y (z) mesh
ZINTS [KM]	Number of fine meshes in each coarse z (θ) mesh
ZONES [IM;JM*KM]	Zone number[b] for each coarse mesh. This array defines the geometric zones to which cross-section materials are assigned via the ASSIGN input array of Block-IV. The zone number must not be greater than NZONE.
LEVELS [IM;JM*KM]	Level indicators for the Block AMR solver. REQUIRED for TRNSOL=3 only, otherwise ignored. The maximum level difference between any two adjacent blocks may be no greater than one. The fine mesh size for each block will be set to 2^L, where L is the input level (greater than or equal to zero). The input fine mesh must be sized such that the number of fine meshes in each coarse mesh is a power of two, and greater than or equal to the assigned level.

a. Definitions of coarse mesh, fine mesh, and zone are given in the chapter "DETAILS OF THE BLOCK-I, GEOMETRY, AND SOLVER INPUT" starting on page 3-1. See note on units on page 2-35. The information entered in this block is written to the CCCC standard interface file GEODST.

b. A zone number of zero indicates the mesh contains a void, and no cross section will be associated with that mesh. The zero zone number is not counted in the total zone count NZONE.

Block-III Details: Nuclear Data

Nuclear Data Type and Options {Required}

Name	Comments
LIB	Name[a] and form of the cross-section data file. Enter as a data item one of the following words:

Word	Description
ISOTXS[b]	CCCC standard isotope ordered binary cross-section file.
XSLIB	ASCII text library supplied in a separate file named XSLIB.
ODNINP	ASCII text library follows after this block of input (after the T of Block-III).
GRUPXS[c]	CCCC standard group ordered cross-section file.
BXSLIB	Binary library supplied as a separate file named BXSLIB. [See "Binary Form of Card-Image Libraries (the BXSLIB file)" on page 6-12.
MACRXS[d]	Use existing files named MACRXS for SOLVER module, SNXEDT for EDIT module. These files were created in a previous run. Under this option, any remaining Block-III input and, unless otherwise specified in Block-I, any PREMIX and MATLS input in Block-IV will be ignored.
XSLIBB	See "XSLIBB Card-Image Library File" on page 6-13.
MACBCD	ASCII form of MACRXS file.
MENDF	(LANL only) See "The Los Alamos MENDF Cross-Section Libraries" on page 6-14.
MENDFG	(LANL only) See "The Los Alamos MENDF5G Gamma Cross-Section Library" on page 6-15.
NDILIB	(LANL only) NDI Library.
other	If a word other than those listed above is entered, the code will use the file with that word as its name, provided that file exists in the users file space. Such a file must be structured as an XSLIB file.

Nuclear Data Type and Options (Cont.) {Required}

Name	Comments
WRITMXS {optional}	Controls the code's writing certain ASCII cross-section files.[e] Enter one of the following words: Word — Description *MACBCD* — Creates the cross-section file named MACBCD, an ASCII image of the MACRXS binary file. *XSLIBB* — Creates the cross-section file named XSLIBB, an ASCII image of the BXSLIB binary file. *XSLIBE* — Creates the cross-section file named XSLIBE, an ASCII file derived from, and corresponding to, the MACRXS binary file. XSLIBE is in Los Alamos 6E12 format (IFIDO=0). *XSLIBF* — Creates the cross-section file named XSLIBF, an ASCII file derived from, and corresponding to, the MACRXS binary file. XSLIBF is in FIDO fixed-field format (IFIDO=1).
LNG {optional}	Number of the last neutron group in a coupled neutron-photon library. Used only to separate neutrons from γs in the edits.
BALXS {optional}	cross-section balance control. Enter one of the following values: WARNING See page 6-23 before using! Value — Description *-1* — balance cross sections by adjusting absorption cross section. *0* — do not balance cross sections. (default) *1* — balance cross sections by adjusting self-scattering cross section.
NTICHI {optional}	MENDF fission fraction to be used for the problem (LANL only). *1/2/3* = Pu239/U235/U238 (default is U235). Will be overridden by any CHIVEC input described below or by any zone-dependent CHI in input Block-V.
CHIVEC [NGROUP] {optional}	Chi vector (fission fraction born into each group). Used for every isotope. Will be overridden by any zone dependent CHI input in Block-V.
CHIMIX	Chi mixing in each zone. *0/1/2* = none/prompt/total. REQUIRES LIB = NDILIB.

a. On UNIX systems, the user may specify a search path for some of these files using the environment variable SNXSPATH. See “Library Search Path” on page 2-108 for details.

b. The CCCC standard for file ISOTXS does not allow the inclusion of the 2L+1 term in the higher order scattering cross section. However, if you have a nonstandard file which contains the 2L+1 term, you may override by setting I2LP1=1. See “Text Cross-Section Library Format” on page 2-47. PARTISN will then convert the cross sections to the appropriate internal form.

c. The 2L+1 term on GRUPXS is treated the same as for ISOTXS. See footnote b.

d. In the convention used in this users guide, a MACRXS library contains “material” cross sections; all the other libraries contain “isotope” cross sections.

e. See “COUPLED NEUTRON-GAMMA CROSS SECTIONS” on page 6-17.

Alternate Library Name {Optional}

Name	Comments
LIBNAME	Alternate name of the library file. May be used only with certain types of libraries. See Table 2.1.

The entries in the LIB input variable normally dictate both the form and the name of the cross-section library. If the user specified ISOTXS, for example, the code would look for a file named ISOTXS and expect it to be in the CCCC format for an ISOTXS file.

For some libraries, the user may specify the form in the LIB array and specify separately the name in the LIBNAME array. The libraries that can be treated this way are shown in Table 2.1.

Table 2.1 LIBNAME Availability

LIB	LIBNAME AVAILABLE?
MACRXS	No
GRUPXS	Yes
ISOTXS	Yes
BXSLIB	Yes
ODNINP	No
MACBCD	No
XSLIBB	No
MENDF[a]	Yes
MENDFG[b]	Yes
XSLIB	Yes
NDILIB[a,b]	Yes
other	Ignored

a. Available only at Los Alamos.

b. MENDF5 or MENDF6 only.

The BXSLIB file requires special treatment. It is normally created when the original library is a text library in the ODNINP or XSLIB form. In subsequent runs, this binary BXSLIB file may be used as the source of the cross-section data. The user may wish to save this file under another name. The program, in future runs, may then access the library for reading by using LIBNAME to specify that name.

This is a wise procedure because some cases using the BXSLIB form as input also require rewriting it in order to add new information. When this situation arises, the rewritten file is always named BXSLIB. Thus, if the original BXSLIB form library had a different name, it would be protected from being overwritten. For the remainder of the current run, the program will access the file named BXSLIB.

Text Cross-Section Library Format {Required if LIB= XSLIB or LIB=ODNINP}

Name	Comments
MAXORD	Highest Legendre order in the scattering tables.
IHM	Number of positions (entries) in each row of the cross-section table.
IHT	Position number of the total cross section.
IHS {optional}	Position number of the self-scatter cross section. (default = IHT + 1).
IFIDO {optional}	Format of the cross-section library. *-1/0/1/2* = Precision(4E18)/Los Alamos(6E12)/fixed-field FIDO/free-field.
ITITL {optional}	A title line precedes each table. *0/1* = no/yes
I2LP1 {optional}	Higher order scattering cross sections on the library contain the 2L+1 term. *0/1* = no/yes. Note: For a non-standard ISOTXS or GRUPXS that contains the 2L+1 term, enter a 1 here.
SAVBXS {optional}	Save the binary form of the ASCII text library XSLIB or ODNINP for use in a subsequent run. Saved on file BXSLIB. *0/1* = no/yes.
KWIKRD {optional}	Process fixed-field FIDO-format, ASCII text library with fast processor at the sacrifice of error checking? *0/1* = no/yes (default=yes).
NAMES [NISO] {optional}	Character name for each of the input isotopes. Can be used later in mixes. (default names are: ISO1, ISO2, . .. etc.).
EDNAME [IHT-3] {optional}	Character name for each of the EDIT cross-section positions used in the cross-section edits. These are the positions before the absorption cross section in the cross-section table. (default names are: EDIT1, EDIT2,...etc.).
NTPI [NISO] {optional}	Number of Legendre scattering orders for each isotope in the library. (default=MAXORD+1 in all positions).
VEL [NGROUP] {optional}	Speeds for each group. Needed only for α calculations.
EBOUND [NGROUP+1] {optional}	Energy group boundaries.

ASCII text libraries may be entered in one of the four forms indicated by the IFIDO input. All four forms share the following features: Cross sections are entered in a table optionally preceded by a title line. A table consists of NGROUP rows of entries. Each row contains the cross sections for a single group and consists of IHM entries. The user specifies the positions in the row occupied by the total and self-scattering cross sections. Order within a row (e.g., for group g) is then as follows:

$$\ldots \sigma_{abs},\quad \nu\sigma_{f},\quad \sigma_{total},\ \ldots \sigma_{g+2\rightarrow g},\ \sigma_{g+1\rightarrow g},\ \sigma_{g\rightarrow g},\ \sigma_{g-1\rightarrow g},\quad \sigma_{g-2\rightarrow g},\quad \text{etc.}$$

Notice that all terms in the scattering matrix are in positions relative to that of the self-scattering position and the rest of the cross sections are in positions relative to the position of the total cross section. The positions before the absorption cross section are frequently used for edit cross sections. For more detail, see “Ordering of Cross Sections Within a Cross-Section Table” on page 6-10.

Different Legendre orders are in different tables, which follow in order.

The user may order the group structure either by increasing energy or by decreasing energy. However, it is conventional and desirable for most problems to order it by decreasing energy, that is, group one is the highest energy. In that case, the scattering cross sections to the left of $\sigma_{g\rightarrow g}$ such as $\sigma_{g+1\rightarrow g}$ are upscattering terms and the terms to the right of $\sigma_{g\rightarrow g}$ are the downscattering terms.

In the Los Alamos format, the table is entered with a standard Fortran 6E12 format.

For greater precision in your input, use the 4E18 option.

In the fixed field FIDO format that ANISN[17] uses, entries are made in six twelve-column fields. Each twelve-column field is divided into three subfields, a two-column numeric field, a one-column character field, and a nine-column numeric field. See page 5-19 for details if you are not familiar with this input. The last field in each table must have the character T in the character position. No array identifier should be used. This format also restricts the usable input operators to T, *, R, -, +, and Z.

In the free field form, entries do not have to be in designated columns. Rather, the rules specified in the chapter “FREE FIELD INPUT REFERENCE” starting on page 5-1 apply. Each table in this form is also terminated with the character T. No array identifier (i.e., array name with appended equals sign) should be used.

Block-IV Details: Cross-Section Mixing

A short summary of the primary mixing arrays, MATLS and ASSIGN, is given here for quick reference. Normally, THESE TWO ARRAYS ARE REQUIRED and, in most problems, would be the only arrays in this block. Other mixing arrays are also briefly described.

There are actually several nested levels of mixing. Each level has the job of calculating values from expressions of the form: $\Sigma_g = \sum_{i=1}^{\kappa} N_i \sigma_{i,g}$ for each group, *g*. The users job is to input the N_i for all the *k* components of the mixture and to specify each component, i. Component *i* has the cross section, $\sigma_{i,g}$. In common usage, for the first level of mixing, $\sigma_{i,g}$ is the effective microscopic cross section, and N_i is the atom density of isotope *i*, and Σ_g is then the macroscopic cross section of some material. In a higher level of mixing, these materials may be homogenized into a single material by using their volume fractions for the N_i. With several nested levels, the user has a great deal of flexibility in defining what Σ_g is for that level. A more complete discussion of mixing will be found in the chapter "MATERIAL MIXING TUTORIAL" starting on page 7-1.

A discussion of cross section processing is outside the scope of this document, but it should be noted that the user needs to be aware of the processing that is inherent in the input library. For instance, for materials in which there are isotopes with cross-section resonances, self shielding of the cross sections for these isotopes may be important and this effect must have been considered in the preparation of the "effective" microscopic cross sections for these isotopes. Since the self shielding is dependent on the amounts and types of the other isotopes in the material, the "effective" cross section is strictly valid only for use in a mixture which has the same composition as was used in the self shielding calculation. If the user desires to use this same "effective" microscopic cross section in some other composition (mix) of material, it is up to the user to verify the accuracy of this approach.

Primary Mixing Arrays {Required}

Name	Description
MATLS[a] [-;MT]	Instructions for mixing "isotopes" or premixes into "materials." See details below.
ASSIGN[b] [-;NZONE]	Assignments of materials to geometric zones. See below.
PREMIX [-;-] {optional}	Instructions for mixing "isotopes" into premixes. See below.

a. The information entered in the MATLS array is written to the CCCC standard interface files NDX-SRF and ZNATDN.

b. Information entered in the ASSIGN array is written to the code-dependent interface file ASGMAT.

In order to understand how cross sections are mixed and the resultant material placed in the problem, we first need a little conceptual information.

The key entities used in specifying the cross-section spatial distribution are coarse mesh, zone, isotope, and material.

The basic geometry of the problem is defined with the coarse meshes specified in Block-II. The geometric areas called zones are also defined there using the ZONES array; the ZONES array designates the zone number assigned to each coarse mesh.

Here in Block-IV, we mix cross sections and assign them to the zones created in Block-II. For the purposes of this discussion, the cross sections found on the input library belong, by definition, to "isotopes," no matter what their true nature. These "isotopes" may then be mixed to form materials, using the MATLS array. Materials are then assigned to zones using the ASSIGN array.

MATLS input array

The general form of a MATLS mix instruction is shown below:

$$\text{MATLS=}\ \ mat_1\ \ comp_1\ den_1,\ \ comp_2\ den_2, \text{ ...etc.... ;}$$

where mat_1 is the desired character name of the first material and $comp_1$, $comp_2$, and so on are the character names of its components which have "densities" of, respectively, den_1, den_2, and so on. Additional materials (i.e., mat_2, mat_3, and so on up to the required

number, MT) are defined in subsequent strings. Each string may contain as many components as necessary (actual limit = 500). A component is usually an isotope from the library, but may also be a temporary material created by the PREMIX array (see below). When the component is an isotope, the den_i is commonly the atom density of the isotope in that material although other definitions exist (See MATSPEC on page 2-54).

Short form: MATLS= *ISOS*

This form specifies that there should be as many materials as isotopes and that isotope number 1 is to be used for material number 1, isotope number 2 is to be used for material number 2, and so on.

In the special case where there is only a single component in a material and its density is unity, the density entry may be omitted as in the first material below:

MATLS= mat_1 $comp_1$; mat_2 $comp_2$ den_2; ...etc.... ;

ASSIGN input array

The general form of the ASSIGN instruction is shown below:

ASSIGN= $zone_1$ mat_1 vol_1, mat_2 vol_2, ...etc.... ;

where $zone_1$ is the desired character name to be used for the first zone (the one specified with numeral 1 in the ZONES array). mat_1, mat_2, and so on are the character names of the materials that will be present in this zone with, respectively, the "volume fractions" vol_1, vol_2, and so on. Additional zones (i.e., $zone_2$, $zone_3$, and so on up to the required number, NZONE) are defined in subsequent strings. Although it is highly recommended that you use character names, here it is convenient to use the numeral for the zone name because it is the same numeral entered in the ZONES array.

Short form: ASSIGN= *MATLS*

This form specifies that there are as many zones as there are materials, and that material number 1 is to be assigned to zone number 1, material number 2 to zone number 2, and so on.

NOTE: The short form ASSIGN=*MATLS* can not be used if you intend to use the ASGMOD input array described later in this section.

PREMIX input array

The PREMIX array forms temporary materials in a way exactly analogous to the way that permanent materials are formed in the MATLS array. The difference in treatment is

that the temporary materials created by PREMIX exist only long enough to complete the mixing; they are not available for assignment to geometric zones, nor are they available for use in material edits.

The general form of a PREMIX mix instruction is shown below:

PREMIX= $tmat_1$ $comp_1$ den_1, $comp_2$ den_2, ...etc.... ;

where $tmat_1$ is the character name of the first material and $comp_1$, $comp_2$, and so on are the character names of its components which have "densities" of, respectively, den_1, den_2, and so on. Additional temporary materials (i.e., tmat_2, $tmat_3$, and so on) may be defined in subsequent strings. A component may be either an isotope from the library or another temporary material created by PREMIX.

The PREMIX array is useful for organizing the mixing input. For instance, it is frequently useful to mix the cross sections for a molecule of water and then in subsequent mix instructions, to input the molecular density of water as opposed to entering the atom density for both hydrogen and oxygen. Other examples are to form average cross sections for an element composed of many isotopes, or to form full density materials and then in later mix instructions to put in the volume fraction of the full density material.

Character Names vs. Numeric Names

In the foregoing discussion, isotopes, materials, and zones were identified by their character names. Optionally, they may be referred to by their ordinal number. Thus, 2 for an isotope name would call for the second isotope on the library. However, this practice is NOT recommended.

THE CHARACTER NAME FORM IS HIGHLY RECOMMENDED. It provides the most straightforward, most readable form. If the character name form is used, the naming input arrays in the table "Miscellaneous Mixing Input" on page 2-54 are not needed.

Using the character name form in one array and the numeric name form in another array is particularly discouraged. However, should one wish to use the numeric form in the MATLS and/or ASSIGN arrays, and then subsequently associate character names with the ordinal numbers, one can use the naming arrays in the following table to do so. This situation could arise if, for some reason, one wanted to use material numbers in the MATLS array, but use character material names in the ASSIGN array.

When the library is of the MENDF form, the character names that must be used for the isotope names are discussed in "The Los Alamos MENDF Cross-Section Libraries" on page 6-14.

Mixing Array for a Concentration Search {Optional}

Name	Description
ASGMOD[a] [-;-]	C_1 parameters used in concentration searches. See the discussion below.

a. The information entered in the ASGMOD array is written to the ASGMAT file together with the information from the ASSIGN and CMOD arrays.

ASGMOD input array

The ASGMOD array is used in conjunction with the ASSIGN array when one wishes to vary the composition of a zone or zones in order to achieve a certain value of k_{eff} or α (i.e., in a concentration search). The concentration (or volume fraction) of material x in zone z is given by the following expression:

$$C(z,x) = C_0(z,x) + C_1(z,x)*CMOD$$

where $C_0(z,x)$ is the base concentration of material x in zone z. This is the concentration (or volume fraction) entered in the ASSIGN array for material x. In these arrays, x is not any kind of an index; correspondence is made by name, rather than by position within the array. Thus, for instance, in a problem that had ten materials, we might only assign one of them to a given zone. It would then probably be in the first position in the ASSIGN array string for that zone even though it might have been say, sixth in the list of all materials.

$C_1(z,x)$ is the corresponding entry in the ASGMOD array for material x in zone z.

CMOD is the search parameter (sometimes called search eigenvalue) that will be varied by PARTISN in order to achieve the desired k_{eff} or α value. In a search calculation, the initial value for CMOD will be the input value EV.

The general form of the ASGMOD instruction is shown below:

ASGMOD= *zone* mat_m vol_m, mat_n vol_n, ...etc.... ;

where *zone* is the character name of any zone in the problem, mat_m, mat_n, and so on are the character names of any of the materials that will be present in this zone, and vol_m,

vol_n, and so on are the C_1 values for respectively, mat_m, mat_n, and so on. Additional zones may be specified in subsequent strings. All zones do not have to appear in the ASGMOD array nor do all materials within a zone have to appear in the string for that zone.

Concentration Modifier {Optional}

Name	Description
CMOD	Concentration modifier. Input value is not used in a search. See the discussion below.

The concentration modifier, CMOD, is varied by PARTISN during a search calculation. For any other type of calculation, a value of CMOD may be input and the composition of the zones will be calculated using the expression above for C(z,x).

Miscellaneous Mixing Input {Optional}

Name	Comments
MATNAM [MT]	Character material names for Materials. Used only if the mat_1 name used in the MATLS array was integer. First entry in MATNAM array is the desired character name for Material number 1, second entry is the desired character name for Material number 2, etc.
ZONNAM [NZONE]	Character zone names for Zones. Used only if the zone name entry in the ASSIGN or ASGMOD array was integer. First entry in the ZONNAM array is the desired character name for Zone number 1, second entry is the desired character name for Zone number 2, etc.

Miscellaneous Mixing Input (Cont.)
{Optional}

Name	Comments
MATSPEC [≤MT]	Tells code whether material mixing in the MATLS array is in terms of atomic densities, atomic fractions, and/or weight fractions. Allowable entries are the words: *ATDENS* (default) atomic densities *ATFRAC*[a] atomic fractions *WTFRAC* weight fractions Can be input as a vector with up to MT entries (one for each Material) [See "Using Atomic Fractions or Weight Fractions (MATSPEC)" on page 7-13.] If less than MT entries are made, the last entry will be used to fill out the array to a length of MT.
ATWT [≤2*NISO] {required[b]}	Atomic weights of the isotopes. If using MATSPEC=ATFRAC or WTFRAC, atomic weights must be available to the code. Entries for the ATWT array are made in pairs, as follows: ATWT= $iso_1\ atwt_1\ \ iso_2\ atwt_2$ .... where iso_n is the isotope name (identifier) for isotope n on the cross-section library and $atwt_n$ is that isotope's atomic weight. [See "Using Atomic Fractions or Weight Fractions (MATSPEC)" on page 7-13].

a. ATFRAC and WTFRAC cannot be used with PREMIX.

b. Required iff MATSPEC=ATFRAC or WTFRAC and atomic weights are not available from the input library.

Block-V Details: Solver Input

Desired Calculation
{Required}

Name	Comments
IEVT	Calculation type: Enter one of the following values:. Value — Description *0* — source *1* — k_{eff} *2* — α (time absorption) search *3* — concentration search *4* — dimension search
ISCT	Legendre order of scattering (default = 0).
ITH	*0/1* = direct/adjoint calculation (default = 0).
IBL	Left boundary condition[a]: *0/1/3* = vacuum/reflective/white (default = vacuum).
IBR	Right boundary condition: *0/1/3* = vacuum/reflective/white (default = vacuum).
IBT	Top boundary condition: *0/1/2/3* = vacuum/reflective/periodic/ white. (default = vacuum).
IBB	Bottom boundary condition: *0/1/2/3* = vacuum/reflective/periodic/ white. (default = vacuum).
IBFRNT	Front boundary condition: *0/1/2/3* = vacuum/reflective/periodic/ white. (default = vacuum).
IBBACK	Back boundary condition: *0/1/2/3* = vacuum/reflective/periodic/ white. (default = vacuum).

a. The left boundary condition applies only for slab, two-angle slab, x-y or x-y-z geometry.

Iteration Controls
{Required}

Name	Comments
EPSI	Convergence precision (default=0.0001).
IITL	Maximum number of inner iterations per group at first (default chosen by code).
IITM	Maximum number of inners allowed when near fission source convergence (default chosen by code).
OITM	Maximum number of outer iterations (default=20).
NOSIGF[a]	Inhibit fission multiplication in a fixed source problem. *0/1* = no/yes.

a. The situation envisioned by this option is that a fission problem has previously been run in which case a FIXSRC file will have been automatically written. That FIXSRC file will contain the pointwise source given by $(1/k_{eff})\chi_g\Phi$ where Φ is the fission distribution. Then the problem is rerun with NOSIGF=1 and INSORS=1 to achieve the same answer as the original. This can then be used to study differences from the original system that do not impact the fission source.

Acceleration Controls
{Optional}

Name	Comments
SRCACC	Choice of transport source acceleration method. Enter one of the following (default is DSA): Word — Description *DSA* — Use diffusion synthetic acceleration. *TSA* — Use transport synthetic acceleration. *NO* — Use no source acceleration.
DIFFSOL	Choice of diffusion operator solver. Enter one of the following words (default is MG for single processor and CG1L for multi-processor): Word — Description *MG* — Use the multigrid solver with 3 line relaxation. *CG3L* — Use conjugate gradient preconditioned by 3 line relaxation. *CG1L* — Use conjugate gradient preconditioned by 1 line relaxation *MGIL* — Use the multigrid solver with 1 line relaxation.

Acceleration Controls
{Optional}

Name	Comments
	MGPLNZ Use the plane-Z multigrid solver. *RBPT* Red black point preconditioner.
TSASN	SN order used for the low order TSA sweeps. Default is 2.
TSAEPSI	Convergence criteria for TSA sweeps. Default is 0.01.
TSAITS	Maximum number of TSA iterations. No default, except that while the fission source is not near convergence, TSAITS is set to 3.
TSABETA	Amount of reduction in scattering cross section for TSA. Default 0.0.

K-Code Convergence
{Optional}

Name	Comments
KCALC	Special Criticality Convergence Scheme. *0/1* = no/yes.

A special convergence scheme may be invoked for problems which require a good eigenvalue, but do not require tight convergence of the pointwise fluxes. It consists of converging the eigenvalue, but not the pointwise fluxes. Normally both must be converged. It also sets the default for eigenvalue convergence to 0.001 rather than 0.0001. To invoke this option to save running time, set the input parameter KCALC to unity.

Output Controls
{Optional}

Name	Comments
FLUXP	Final flux print. *0/1/2* = no/isotropic/all moments.
XSECTP	Cross-section print. *0/1/2* = no/principal/all.
FISSRP	Fission source rate print. *0/1* = no/yes.
SOURCP	Source print. *0/1/2/3* = no/as input/normalized/both.
ANGP	Print angular flux. *0/1* = no/yes. CAUTION! This is very LARGE output. ANGP=1 will cause the RAFLXM or AAFLXM file to be written. Only valid for TRNSOL=0.
BALP	Coarse Mesh Interval Print Options. Enter one of the following values: Value — Description *0* — None *1* — Print coarse mesh balance tables.
RAFLUX	Prepare angular flux file (RAFLXM or AAFLXM). *0/1* = no/yes.
RMFLUX	Prepare flux moments file (RMFLUX or AMFLUX). *0/1* = no/yes.
ASLEFT	Write right-going angular edge flux at plane *i* to file bsleft.
ASRITE	Write left-going angular edge flux at plane *i* to file bsrite.
ASBOTT	Write top-going angular edge flux at plane *j* to file bsbot.
ASTOP	Write bottom-going angular edge flux at plane *j* to file bstop.

Output Controls (Cont.)
{Optional}

Name	Comments
ASFRNT	Write back-going angular edge flux at plane *k* to file bsfrnt.
ASBACK	Write front-going angular edge flux at plane *k* to file bsback.

Miscellaneous Solver Input {Optional}

Name	Comments
LSSN	Use the LSSN positive scattering method. *0/1*=no/yes
LSXS	Write/read the modified LSSN moments from the file lsxsdat. *0/1/2*=no/write/read
TRCOR	Apply transport correction[a] to cross sections on MACRXS file. Enter one of the following words: Word — Description *DIAG* — Use diagonal transport correction. *BHS* — Use Bell-Hansen-Sandmeier correction. *CESARO* — Use Cesaro "correction." *NO* — (or omit entry) Use no correction.
BHGT	Buckling height (in cm. if macroscopic cross section in cm^{-1}.) Used only for plane, cylindrical, and two-angle plane geometries. (default=0.0, which is treated as infinity).
BWTH	Buckling width. Used only for plane and two-angle plane geometries. (default=0.0, which is treated as infinity).
NORM	Normalize the fission source rate to this value when IEVT.GE.1 or normalize the inhomogeneous source rate to this value when IEVT=0. NORM=0 means no normalization. (Integral of source rate over all angle, space, and energy = NORM, except for k_{eff} problems where the integral is equal to NORM*k_{eff}.) Any fluxes printed here (i.e., caused by setting FLUXP nonzero) will be normalized consistently with this source rate.
CHI [NGROUP;M]	Fission fraction born into each group.[b] Enter by zone up to M zones. Succeeding zones (i.e., zones M+1 through NZONE) will use the CHI values from zone M.
DEN [IT;JT*KT] or	Density factor to use at each fine mesh point.
DENX [IT][c] and/or	Density factor to use at each fine x-mesh (default=1).

Miscellaneous Solver Input (Cont.)
{Optional}

Name	Comments
DENY [JT] and/or	Density factor to use at each fine y-mesh (default=1).
DENZ [KT]	Density factor to use at each fine z-mesh (default=1).
ANGFLX	Write time-dependent angular fluxes to disk. Not operational for TRNSOL=3. *0/1*=no/yes
FLXMOM	Write flux moments to disk. Not operational for TRNSOL=3. *0/1*=no/yes

a. For more information, see “Transport Corrections for the Cross Sections (TRCOR)” on page 3-35.

b. This input will override any previous CHI from earlier blocks or from any cross-section library which contains CHI.

c. In this second form, the density factor DEN(i,j,k,), at mesh interval (i,j,k) is computed as follows: DEN(i,j,k) = DENX(i)*DENY(j)*DENZ(k).

Quadrature Details {Optional}

Name	Description
GRPSN [NGROUP][a]	S_n order to be used for each group.
WGTDIA	Use weighted diamond angular differencing (1-D spheres and cylinders only). *0/1*=no/yes.
WGT [MM[b]]	Quadrature weights.
MU [MM]	Mu cosines.
ETA [MM]	Eta cosines.
NLL [NN]	Number of angles per level for general quadrature. REQUIRED for IQUAD=3.

a. Values must be less than or equal to ISN in Block-I. The GRPSN option may not be used with boundary source input.

b. MM=ISN*(ISN+2)/8 for iquad = 1, 2, and 5.
MM = (ISN/2)**2 for iquad = 6.

Transport Solver Details {Optional}

Name	Description
TRNSOL	Transport solver type. *0/1/2/3*=vectdl/seqdp/blockj/block AMR. Default is vectdl for single processor and seqdp for multi-processor.
NODAL	Spatial differencing scheme. *0/1/2/3*=standard low-order (DD,DD/STZ,AWDD)(default)/exponential discontinuous/linear discontinuous/linear discontinuous via the exponential discontinuous solver.
WDAMP[NGROUP]	Flags to activate adaptive weighted diamond differencing (AWDD)[18] for each group. 0.0/W = no/activate AWDD with parameter W.[a] If W = 0.0, the default diamond with fixup is used.

a. Recommend $1.0 < W < 4.0$ for shielding applications.

Flux Guess From a File {Optional}

Name	Comments
INFLUX	Read the initial flux guess from a file.[a] *0/1* = no/yes. If ITH=0 and ISCT>0, and the flux moments file RMFLUX exists, read the initial flux guess from RMFLUX. Otherwise, read the initial flux guess from the RTFLUX file. If ITH=1 and ISCT>0, and the adjoint moments file AMFLUX exists, read the initial flux guess from AMFLUX. Otherwise, read the initial flux guess from the ATFLUX file.

a. There is presently no text input flux guess available for PARTISN.

General Eigenvalue Search Control[a] {IEVT >1}

Name	Comments
IPVT	Type of eigenvalue to search for in a concentration or dimension search. *0/1/2* = none / k_{eff} / α. (default = 1).
PV	Value of k_{eff} or α to which to search. (default = 1.0 if IPVT=1, 0.0 if IPVT=2.
EV	Initial search parameter. Value at which to start the search parameter. (default=0).
EVM	Initial search parameter increment. Amount by which to change search parameter in the first step of a search. (default = 0.01 for DSASRCH=0 and 2.5 for DSASRCH=1).
XLAL	Lambda lower limit for search. (default = 0.01).
XLAH	Lambda upper limit for search. (default = 0.5).
XLAX	Lambda convergence criterion for second and subsequent search steps. (default = 10*EPSI).
POD	Parameter oscillation damper. (default=1.0).
DSASRCH	Use DSA estimate of d(alpha) / d(lambda) for alpha eigenvalue search. *0/1* = no/yes.

a. See "Eigenvalue Searches" on page 3-38 for definitions of these quantities.

PARTISN can vary the composition or dimensions of a zone (or zones) in order to achieve a desired k_{eff} or α value. The search input consists of the above general search input plus input specific to the type of search being performed.

Dimension Search Input
{Required if IEVT=4}

Name	Comments
XM [IM]	x-dimension fractional change per coarse mesh.
YM [JM]	y-dimension fractional change per coarse mesh.
ZM [KM]	z-dimension fractional change per coarse mesh.

The dimension search requires the XM and/or YM and/or ZM input as well as the general search input above. During the search, PARTISN varies the search parameter (sometimes called the search eigenvalue) denoted by EV in the following expressions to change the coarse mesh boundaries:

$$XMESH_{i+1} = XMESH_i + \{XMESH_{i+1}-XMESH_i\} * [1.0+EV*XM_i], \quad i=1, \ldots, IM$$

$$YMESH_{j+1} = YMESH_j + \{YMESH_{j+1}-YMESH_j\} * [1.0+EV*YM_j], \quad j=1, \ldots, JM$$

$$ZMESH_{k+1} = ZMESH_k + \{ZMESH_{k+1}-ZMESH_k\} * [1.0+EV*ZM_k], \quad k=1, \ldots, KM$$

Although they may seem a bit awkward at first, the user will find these expressions to be quite flexible. With proper choice of the XM_i, YM_j, and ZM_k values, the user can move any or all of the coarse mesh boundaries while allowing others to remain stationary. The

quantities in { } in the above expressions are always formed from the original input values.

Concentration Search Input {Required if IEVT=3}

Name	Description
The solver input for a concentration search is to set IEVT = 3 (page 2-56) and input the general eigenvalue search controls. But you must also input the ASGMOD[a] array in Block-IV.	

a. A concentration search involves the mixing instructions. A discussion of the ASGMOD array is found in the mixing input description on page 2-53.

Parallelization Details {Optional}

Name	Description
NPEY	Number of processors to be used for the Y-processor mesh.
NPEZ	Number of processors to be used for the Z-processor mesh.
NCHUNK	Specifies the number of X spatial planes to be solved before communicating for TRNSOL=1 (default 10). For TRNSOL=3, specifies the number of X spatial planes to be grouped together during decomposition (default 1).
The numbers of processors used is NPEY*NPEZ. If NPEY*NPEZ is a power-of-two, then entry of NPEY and NPEZ is OPTIONAL. Otherwise, entry of NPEY and NPEZ is REQUIRED.	

Mesh Potential Input {Optional}

Name		Comments
MSHCOL		Mesh potential control.
	0	No Mesh collapse (default).
	1	For single-level problems, print mesh coarsening suggestions to the incol3d file and exit without solving. For Block AMR problems, print final block level suggestions to output and exit without solving.
	2	Regenerate the mesh using the mesh potential information and solve.
MSHCEWT [IGM]		Energy weighting spectrum to be used with the mesh potential function. Default 1.0.
MSHCASG		Weighting parameter for the total cross section (default 0.9).
MSHCBNF		Weighting parameter for the nu*fission cross section (default 0.1).
MSHCRBN		Mesh potential value (default 0.5/NGROUP). For single-level problems, the mesh potential function will attempt to coarsen planes of cells where the calculated mesh potential (based on MSHCEWT, MSHCASG, and MSHCBNF) is less than the lower bound MSHCRBN. For Block AMR, a block's level may be reduced if the maximum potential in any cell is lower than MSHCRBN.
MSHCXVD		For single-level problems, allow coarsening of planes of void cells. For Block AMR, allow void blocks to be reduced to level 0. *0/1*=yes/no.
MSHCXRG[6]		Lower and upper limits of fine mesh cells in each direction between which no coarsening is allowed (ex. 0 10 0 20 0 5). For Block AMR, MSHCXRG should be filled with the lower and upper limits of the coarse mesh region that needs protection from level reduction. All blocks in that region will be protected. The array elements correspond to the (left right bottom top front back) limits. Default 0 0 0 0 0 0.

Mesh Potential Input {Optional}

Name	Comments
MSHCPR	Include mesh potential diagnostic information in the output. Intended to help users override default mesh potential parameters when they do not result in the desired amount of coarsening. For single-level problems, the output is the maximum potential for every plane of cells in every direction. For Block AMR, the output is the maximum potential in the block containing the most important cells and in the block containing the least important cells. *0/1*=no/yes.
MSHCMSS	The maximum relative change in source between neighboring cells that is allowed during coarsening (Maximum Source Slope). A negative value suppresses source control. Default 0.5.

Time-Dependent Input {Optional}

Name	Comments
TIMEDEP	Run in time-dependent mode (DD/STZ in time). *0/1* = no/yes.
T0	Initial time.
TS	Final time.
DELTI	Initial time step size.
DELTMIN	Minimum allowed time step size (default 0.0001).
INITTF	Time-dependent flux initialization. *0/1/2* = given/settle/stepstart.
EOM	The input source (for source problems) is set to this value as a floor (default 0.0).
EFACT	Time step controller parameter. For source only problems, if the change in the maximum pointwise total reaction rate is less than 0.2*EFACT, then the time step is increased by 20%. For fissioning problems, if the change in the total neutron population and the change in the fission source are both less than 0.1*EFACT, then the time step is increased by 20%. Default 0.5.
STIMES [M]	Time points at which the source is assumed linear (i.e., the source has a linear dependence between STIMES(m) and STIMES (m+1)).
SAMP [M]	Relative value of the source amplitude at each source time point.

Time-Dependent Input {Optional}

Name	Comments
XTIMES [N]	Time points at which the cross section is assumed linear (i.e., the cross section has a linear dependence between XTIMES(n) and XTIMES(n+1)).
XAMP [N]	Relative value of the cross section at each XTIMES time point. The cross section change is governed by the ASGMOD array in Block IV of the input.
DTIMES [L]	Time boundaries for dump intervals.
NTIMES [L]	Number of dumps to be (approximately) uniformly spaced within each dump interval.
RDMPNME	Dump name prefix to which the cycle number will be appended.

Volumetric Source Options {Optional}

Name	Comments
INSORS	Read source from interface file FIXSRC. *0/1*=no/yes
---- For a text-input source, choose one of the following options:	
Option 1:	
SOURCE [NGROUP; NMQ]	Source spectrum for each of NMQ[a] moments. (Spatial distribution is assumed to be flat with value unity)
Option 2:	(input all three arrays)
SOURCX [IT;NMQ][b]	x (or r) spatial distribution for each moment.
SOURCY [JT;NMQ]	y (or z) spatial distribution for each moment.
SOURCZ [KT;NMQ]	z (or θ) spatial distribution for each moment.
(Spectrum is assumed to be flat with value unity)	
Option 3:	(input all four arrays)

Volumetric Source Options {Optional}

Name	Comments
SOURCE [NGROUP; NMQ]	Source spectrum.
SOURCX [IT;NMQ]	x (or r) spatial distribution for each moment.
SOURCY [JT;NMQ]	y (or z) spatial distribution for each moment.
SOURCZ [KT; NMQ]	z (or θ) spatial distribution for each moment.
Option 4:	
SOURCF [IT;JT*KT*NGROUP*NMQ]	Spatial distribution for each row, group, and moment.
Option 5:	(input both arrays)
SOURCE [NGROUP; NMQ]	Source spectrum.
SOURCF [IT;JT*KT*NMQ]	Spatial distribution for each row and moment.

a. NMQ is not an input value but is computed from the number of strings read. NMQ must correspond exactly to the number of moments in a P_n expansion of the source. The number of moments is $(n+1)^2$ in 3-D, (n+1)*(n+2)/2 in 2-D, (n+1) for 1-D slabs and spheres, $(n+2)^2/4$ for 1-D cylinders, and $(n+1)^2$ for two-angle slabs. n must be less than or equal to ISCT. See page 8-25 for more details.

b. Only in option 4 is the complete pointwise source array, SOURCEF(i,j,k,g,m), given. In all other cases, it must be formed from the lower dimension arrays that are input. That calculation is done by forming the product of those arrays. Thus, in option 3, where the source spectrum, SOURCE(g,m), and the spatial distributions SOURCX(i,m), SOURCY(j,m), SOURCZ(k,m) are given (for moment m), the full source at mesh point (i,j,k) in group g for moment m is calculated as follows:

SOURCE(i,j,k,g,m) = SOURCE(g,m)*SOURCX(i,m)*SOURCY(j,m)*SOURCZ(k,m)

Boundary Source Input {Optional}

Name	Comments
BSFILE[6] Use boundary source interface file for boundary sources. The six positions correspond to each boundary source file in the following order; BSLEFT BSRITE BSBOT BSTOP BSFRNT BSBACK. For example, BSFILE= 0 1 0 0 1 1 indicates to use BSRITE, BSFRNT and BSBACK boundary source interface files. *0/1*=no/yes	
----- For a text-input source, choose one of the following options:	
Option 1: Isotropic Boundary Source	
SILEFT [NGROUP;JT*KT]	Isotropic source on the left face. (Spectrum at each y,z mesh interval.)
SIRITE [NGROUP;JT*KT]	Isotropic source on the right face.
SIBOTT [NGROUP;IT*KT]	Isotropic source on the bottom face.
SITOP [NGROUP;IT*KT]	Isotropic source on the top face.
SIFRNT [NGROUP;IT*JT]	Isotropic source on the front face.
SIBACK [NGROUP;IT*JT]	Isotropic source on the back face.
Option 2: Full Angular Boundary[a] Source	
SALEFT [MM[b]*4;NGROUP*JT*KT]	Angular flux on the left for each angle, group, and y,z mesh interval.
SARITE [MM*4;NGROUP*JT*KT]	Angular fluxex on the right face.
SABOTT [MM*4;NGROUP*IT*KT]	Angular fluxes on the bottom face.
SATOP [MM*4;NGROUP*IT*KT]	Angular fluxes on the top face.
SAFRNT [MM*4;NGROUP*IT*JT]	Angular fluxes on the front face.
SABACK [MM*4:NGROUP*IT*JT]	Angular fluxes on the back face.

a. The order of the angles is identical to that used in teh S_n Constants table in the output file. The order of the angular octants is: mu < 0, eta < 0, tsi < 0; mu > 0, eta < 0, tsi < 0; mu < 0, eta > 0, tsi < 0: mu > 0, eta > 0, tsi < 0; mu < 0, eta < 0, tsi > 0; mu > 0, eta < 0, tsi > 0; mu < 0, eta > 0, tsi > 0, and mu > 0, eta > 0, tsi > 0, where each angular boundary source requires four octants for specification.

b. See “Quadrature Details” on page 2-63 for value of MM.

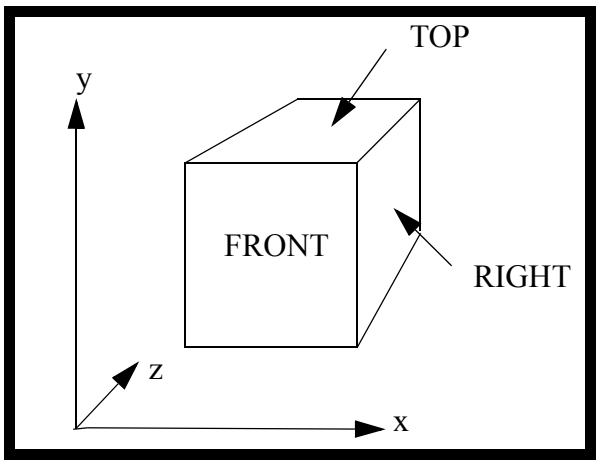


Figure 2.2 Orientation of Faces

Note the non-standard orientation of the z axis and accordingly that top is NOT in the z direction.

Within option 3 are four suboptions that involve different combinations of ‘vector’ type input for the boundary sources. All the vector input is defaulted to unity if not entered explicitly. The full angular source for the left face, for example, is constructed as follows for the 4 options (the source construction on the other faces is analogous):

Option 3a- S(m,g,y,z) = BSLFTG(g)*BSLFTY(y)*BSLFTZ(z)*BSLFTA(m)

Option 3b- S(m,g,y,z) = BSLFTG(g)*BSLFTYZ(y,z)*BSLFTA(m)

Option 3c- S(m,g,y,z) = BSLFTG(g)*BSLFTYZ(y,z,m)

Option 3d- S(m,g,y,z) = BSLFTG(g)*SALEFT(m,y,z)

Boundary Source Vector Input Combinations {Optional}

Left	Right	Bottom	Top	Front	Back
Option 3a: Boundary Source By Product Of Vectors					
BSLFTG [NGROUP]	BSRITG [NGROUP]	BSBOTG [NGROUP]	BSTOPG [NGROUP]	BSFRNG [NGROUP]	BSBAKG [NGROUP]
BSLFTY [JT]	BSRITY [JT]	BSBOTX [IT]	BSTOPX [IT]	BSFRNX [IT]	BSBAKX [IT]
BSLFTZ [KT]	BSRITZ [KT]	BSBOTZ [KT]	BSTOPZ [KT]	BSFRNY [JT]	BSBAKY [JT]
BSLFTA [MM*4]	BSRITA [MM*4]	BSBOTA [MM*4]	BSTOPA [MM*4]	BSFRNA [MM*4]	BSBAKA [MM*4]
Option 3b: Boundary Source By Product Of A Spectrum Vector, An Angular Distribution Vector, And A 2d Spatial Array					
BSLFTG [NGROUP]	BSRITG [NGROUP]	BSBOTG [NGROUP]	BSTOPG [NGROUP]	BSFRNG [NGROUP]	BSBAKG [NGROUP]
BSLFTYZ [JT;KT]	BSRITYZ [JT;KT]	BSBOTXZ [IT;KT]	BSTOPXZ [IT;KT]	BSFRNXY [IT;JT]	BSBAKXY [IT;JT]
BSLFTA [MM*4]	BSRITA [MM*4]	BSBOTA [MM*4]	BSTOPA [MM*4]	BSFRNA [MM*4]	BSBAKA [MM*4]
Option 3c: Boundary Source By Product Of A Spectrum Vector And A 3d Space-Angle Array					
BSLFTG [NGROUP]	BSRITG [NGROUP]	BSBOTG [NGROUP]	BSTOPG [NGROUP]	BSFRNG [NGROUP]	BSBAKG [NGROUP]
BSLFTYZ [JT;KT* MM*4]	BSRITYZ [JT;KT* MM*4]	BSBOTXZ [IT;KT* MM*4]	BSTOPXZ [IT;KT* MM*4]	BSFRNXY [IT;JT* MM*4]	BSBAKXY [IT;JT* MM*4]
Option 3d: Boundary Source By Product Of A Spectrum Vector And A 3d Angle-Space Array					
BSLFTG [NGROUP]	BSRITG [NGROUP]	BSBOTG [NGROUP]	BSTOPG [NGROUP]	BSFRNG [NGROUP]	BSBAKG [NGROUP]
SALEFT [MM*4; JT*KT]	SARITT [MM*4; JT*KT]	SABOTT [MM*4; IT*KT]	SATOPT [MM*4; IT*KT]	SAFRNT [MM*4; IT*JT]	SABAKT [MM*4; IT*JT]

First Collision Source Input {Optional}

Name	Comments
FCSRC	Use First Collision Source Option. Only the PTANA and UMCFLUX options are available in parallel. Enter one of the following words: Word — Description *PTANA*[a] — Use the semi-analytic point source at position FCXPOS, FCYPOS, FCZPOS given below. x-y-z geometry only. *RAYTR*[b] — Use the ray tracing first collision source. x-y, r-z, x-y-z, or r-z-θ geometry only. *PTRAY* — Use ray trace point source as in *PTANA*. *PTBEAML* — Use the point beam option on the left face of the geometry at position FCXPOS,FCZPOS; polar angle= FCPOANG and azimuthal angle= FCAZANG. A beam incident on other faces is entered the same way except the last letter is changed to R,T,B,F,K for the right, top, bottom, front, back faces respectively. *UMCFLUX*[c] — Use the "Unde McFlux" ray tracing first collision source. x-y or x-y-z geometry only. *NO* — (or omit entry) - don't use
FCNRAY	Number of ray tracings/batch to use with ray tracing first collision source. (default = 10000).
FCNTR	Number of batches (trials) to use with ray tracing first collision source. If FCNTR is negative, read the restart file UCFLUX, and perform |FCNTR| additional batches of ray traces (default = 25). If this is greater than 1 and the point beam option is used, then a second collision source is done using ray tracing from the first collision source.[d]

First Collision Source Input {Optional}

Name	Comments
FCRSTRT	If FCRSTRT=0, generate the ucflux file (single-PE only) and solve. For FCRSTRT=1, read in an existing ucflux file, perform FCNTR additional trials, then solve. If FCNTR=0 and FCRSTRT=1, then read in the existing ucflux file and solve without performing additional trials. For FCRSTRT=2, return immediately after generating the ucflux file. This allows a ucflux file to be generated on 1 PE (FCRSTRT=2), then restarted and solved on multi-PE's (FCRSTRT=1 and FCNTR=0). If ucflux already exists and FCRSTRT=2, read in the existing ucflux file, perform FCNTR additional trials, then exit. *0/1/2*=no/yes/generate the ucflux file.
FCWCO	Weight cutoff for the ray tracing first collision source. When a ray has been traced a sufficient distance to attenuate its weight to less than wtmin * FCWCO, where wtmin is the minimum source starting weight, then that ray is terminated. Default entry is 0.0, which corresponds to a FCWCO value of EPSI**2.
FCXPOS	X position of the point source for all point source options.
FCYPOS	Y position of the point source for all point source options.
FCZPOS	Z position of the point source for all point source options.
FCPOANG	Polar angle in radians of the point beam source.[e]
FCAZANG	Azimuthal angle in radians of the point beam source.[f]

a. The point source will be moved within the mesh cell to the left, bottom, front corner of the cell for accuracy considerations.

b. The ray tracing first collision source may be used with any of the volumetric or boundary source input options.

c. The "Unde McFlux" ray tracing first collision source may be used with volumetric source input only.

d. If FCNTR is greater than 1, the batches will be used to provide an estimate of the maximum relative statistical error in the calculated uncollided flux for every k plane.

e. The polar angle is measured from the positive Z axis.

f. The azimuthal angle is measured from the positive X axis in the XY plane.

Block-VI Details: Edit Input

The input* in this block controls the calculation of reaction rates using the flux solution from the solver module.

Edit Spatial Specifications {Required[a]}

Name	Comments
PTED	Do edits by fine mesh. *0/1* = no/yes.
ZNED	Do edits by zone. *0/1* = no/yes. (i.e., edit zone, not SOLVER zone. See EDZONE input below.)
POINTS [≤IT*JT*KT] {optional}	Fine mesh point (or interval) numbers at which point edits are desired. USED ONLY IF PTED=1. (Default= all points)
EDZONE [IT;JT*KT] {optional}	Edit zone number for each fine mesh interval. USED ONLY IF ZNED=1. (default= SOLVER coarse mesh interval numbers, see ZONES array, Block-II on page 2-41)

a. Either PTED or ZNED or both must be unity in order to produce reaction rate edits.

* More details for the input for edits are given in the chapter "RUNNING THE EDIT MODULE" starting on page 4-1.

Reaction Rates from Cross Sections[a] {Optional[b]}

Name	Comments
EDXS [≤NEDT] {required[c]}	Cross-section types to be used in forming reaction rates. May be entered by integer (denoting edit position of desired cross-section type) or by the character name of the cross-section type. See the table"Edit Cross-Section Types by Position and Name" on page 2-81 or "MENDF Library Edit Cross Sections" on page 2-88 for the available names. NEDT is the total number of edit cross-section types available from the input cross-section library. (default = all shown in the table) Note: The cross-section types specified in this array apply to any or all of the following edit forms: RESDNT, EDISOS, EDCONS, EDMATS.
RESDNT	Do edits using the resident macroscopic cross section at each point. *0/1* = no/yes.
EDISOS [≤NISO]	Character names of the isotopes to be used in forming Isotopic reaction rates. The ordinal number may alternately be used but is not recommended. (default = none).
EDCONS [≤NISO]	Character names of the isotopes to be used in forming resident Constituent (partial macroscopic) reaction rates. The ordinal number may alternately be used but is not recommended. (default = none).
EDMATS [≤MT]	Character names of materials to be used in forming Material (macroscopic) reaction rates. The ordinal number may alternately be used, but is not recommended. (default = none).
XDF[d] [IT] YDF [JT] ZDF [KT]	Fine mesh density factors for the x(or r), y(or z) and z(or θ) directions, respectively. The density factor is used to multiply resident Constituent (see EDCONS), resident macroscopic (see EDMATS), and resident macroscopic (see RESDNT) reaction rates only. (Default= all values unity).

a. See chapter "RUNNING THE EDIT MODULE" starting on page 4-1 for further discussion.

b. But either something in this grouping or the next must be input in order to produce reaction rate edits.

c. You must also enter one or more of the arrays EDISOS, EDCONS, EDMATS, or RESDNT.

d. If density factors were used in SOLVER to modify the cross sections at each mesh interval, the same density factors must be provided here in the XDF and/or YDF and/or ZDF arrays as well. The density factor at mesh interval (i,j,k) is computed as:

XDF(i)*YDF(j)*ZDF(k)

Edit Cross-Section Types by Position and Name

CROSS-SECTION INPUT VIA ISOTXS or GRUPXS			CROSS-SECTION INPUT VIA ASCII TEXT		
Type	EDIT Position	Name[a]	Type	EDIT Position	Name
chi	1	CHI.....	not used	1	CHI.....
nu-fission	2	NUSIGF..	nu-fission	2	NUSIGF..
total	3	TOTAL...	total	3	TOTAL...
absorption	4	ABS.....	absorption	4	ABS.....
(n,p)	5	N-PROT..	1[b]	5	EDIT1...[c]
(n,d)	6	N-DEUT..	2	6	EDIT2...
(n,t)	7	N-TRIT..	3	7	EDIT3...
(n,α)	8	N-ALPH..	.	.	.
(n,2n)	9	N-2N....	.	.	.
(n,γ)	10	N-GAMM..	.	.	.
fission	11	N-FISS..	N=IHT-3	4+N	EDITN...
transport	12	TRNSPT..			

a. Names are eight characters. A period within a name in this table denotes a blank.

b. Denotes position in the cross-section table. All cross sections in positions 1 through IHT-3 in the cross-section library are EDIT cross sections chosen by the user.

c. These are the default names that may be overridden with the user-option names in the EDNAME array of Block-III.

Reaction Rates from User Response Functions {Optional[a]}

Name	Comments
RSFE [NGROUP;M] {required if user input response functions are desired}.	Response function energy distribution for each of the M different response functions desired. The number of different response functions is arbitrary (but must be fewer than 500). Data are entered as M strings, each with NGROUP entries beginning with group 1.
RSFX [IT;M][b]	Response function x (or r) distribution for M functions.
RSFY [JT;M]	Response function y (or z) distribution for M functions.
RSFZ [KT;M]	Response function z (or θ) distribution for M functions.
{optional}	The above data are entered as M strings of IT, JT or KT entries beginning with mesh point 1. (default=1.0)
RSFNAM [M] {optional}	Character names for the user-input response functions specified above. (default = RSFP1, RSFP2,...RSFPM)

a. But either something from this grouping or the previous one must be input in order to produce reaction rate edits.

b. The m-th response function at space point (i,j,k) and energy group g is computed as:

RSFX(i,m) * RSFY(j,m) * RSFZ(k,m) * RSFE(g,m)

Energy Group Collapse Specifications {Optional}

Name	Comments
ICOLL [NBG]	Edit energy group collapsing option: Number of SOLVER energy groups in each EDIT broad group. The NBG entries must sum to NGROUP. (default = 1 energy group per EDIT broad group)
IGRPED	Print option on energy groups. Enter one of the following values: Value — Description *0* — Print energy group totals only *1* — Print broad groups only *2* — Print broad groups only (same as 1) *3* — Print both broad groups and totals

Reaction Rate Summing {Optional}

Name	Comments
MICSUM [≤500 sums]	Cross-section reaction rate summing specifications. The MICSUM array is a packed array with data entered as follows: A set of Isotope numbers or names is given, followed by a set of cross-section type position numbers or names (see "Edit Cross-Section Types by Position and Name" on page 2-81). These sets are delimited with an entry of 0 (zero). Reaction rates are calculated for each Isotope specified for each cross-section type specified and summed to form the first sum. The next two sets of data are used to form the second sum, etc. Up to 500 sums can be specified. (for more detail, see "Response Function Summing Options" on page 4-13).
IRSUMS [≤500 sums]	Response function reaction rate summing specifications. The IRSUMS array is input as follows: A set of response function numbers or names is entered and the set delimited with an entry of 0 (zero). Reaction rates are calculated using these response functions, and the rates are summed to form the first sum. The next set of data is used to form the second sum, etc. Up to 500 sums can be specified. See page 4-13 for more detail.

Mass Inventories {Optional}

Name	Comments
MASSED	Calculate and print mass inventories by zone. *0/1/2/3* = none/ solver zones/edit zones/both (default=1). This option is active only if atomic weights are present. See ATWT on page 2-55.

Power Normalization {Optional}

Name	Comments
POWER {required}	Normalize to POWER megawatts.[a] All printed reaction rates and the fluxes on files RTFLUX and RZFLUX (if requested) will be normalized. Fluxes are normally not printed here in the EDIT module, although they may be extracted by using a unit response function. Any such fluxes will also be normalized to POWER. Contrast the normalization on these printed fluxes to those printed by the FLUXP input in the SOLVER Block (see NORM on page 2-61).
MEVPER {required}	MeV released per fission (default=210 MeV). This value will be used along with the calculated fission rate to determine the power. For the power calculation, PARTISN needs to know which cross section is the fission cross section. It uses the one from the library that has the name N-FISS. If one uses an ISOTXS or GRUPXS library that designation is automatically provided (See "Edit Cross-Section Types by Position and Name" on page 2-81). But if one uses an ASCII text library, either ODNINP or XSLIB, then the name N-FISS must be entered in the proper place in the EDNAME array (page 2-47).

a. Note that this normalization is meaningless if you are using the results of an adjoint run.

Miscellaneous Edit Items {Optional}

Name	Comments
RZFLUX	Write the CCCC standard zone[a] flux file RZFLUX or AZFLUX. *0/1* = no/yes.
RZMFLX	Write the code-dependent zone[b] flux moments file RZMFLX or AZMFLX. *0/1* = no/yes.
EDOUTF[c]	ASCII output files control. Enter one of the following values: Value — Description *-3* — Write both EDTOGX (without scalar fluxes) and EDTOUT files. *-2* — Write EDTOGX file (without scalar fluxes). *0* — Write neither file. (default) *1* — Write EDTOUT file. *2* — Write EDTOGX file (with scalar fluxes). *3* — Write both EDTOGX (with scalar fluxes) and EDTOUT files.
BYVOLP	Printed point reaction rates will have been multiplied by the mesh volume. *0/1* = no/yes.
AJED[d]	Regular (forward) edit/Adjoint edit. Regular edit uses the RTFLUX scalar flux file; adjoint edit uses the ATFLUX flux file. *0/1* = regular/adjoint.
FLUXONE	Flux override. *0/1* = no/yes. Replaces all the input fluxes by unity. Useful for seeing the cross sections used in cross-section edits. WARNING! Meaningful reaction rates cannot be obtained when this switch is on.

a. RZFLUX and AZFLUX are organized by solver zones.

b. RZMFLX and AZMFLX are organized by solver zones.

c. See “ASCII File Output Capabilities (the EDOUTF Parameter)” on page 4-15.

d. See “Adjoint Edits” on page 4-15.

Special Plot Linkage {Optional}

Name	Comments
PRPLTED	Write an ASCII file of the pointwise reaction rates to link to the TECPLOT© plotting package available commercially for a SUN workstation. *0/1/2/3* = print only/nothing/tecplot file/both print and tecplot file.
IPLANE [≤IT]	x mesh numbers at which to write a y-z distribution.
JPLANE [≤JT]	y mesh numbers at which to write a x-z distribution.
KPLANE [≤KT]	z mesh numbers at which to write a x-y distribution.

To exercise this option, the user must have set PTED=1. The code will calculate reaction rates at all the fine mesh intervals, and any POINTS input will be ignored.

To link to the TECPLOT© code, the user chooses option 2 or 3. Separate ASCII files called rsp.dat and med.dat will be written for the response function and material edits, respectively. These files are in input form for the TECPLOT© preprocessor at the planes specified by IPLANE, JPLANE, and/or KPLANE.

If option 0 (print only) is chosen, no TECPLOT© files will be written but the reaction rates will be printed. The format of this printout is organized in a two-dimensional way unlike the normal printout from the EDIT module.

MENDF Library Edit Cross Sections

Reaction Type	Name	Description
χ	CHI	fission spectrum
$\nu\sigma_f$	NUSIGF	effective nu-sigma-fission
σ_t	TOTAL	Total cross section
σ_a	ABS	absorption[a]
(n,n)	MEND1	elastic scattering
(n,n´)	MEND2	inelastic scattering
(n,2n)	MEND3	n,2n scattering
(n,3n)	MEND4	n,3n scattering
(n,γ)	MEND5	γ production
(n,α)	MEND6	α production
(n,p)	MEND7	proton production
(n,f)	MEND8	direct fission
(n,n´)f	MEND9	second-chance fission
(n,2n)f	MEND10	third-chance fission
(n,F)	N-FISS	[(n,F) = (n,f) + (n,n´)f + (n,2n)f]
χ_p	MEND12	prompt fission spectrum (only for fissionable materials)
χ_t	MEND13	total fission spectrum (only for fissionable materials)

a. σ_a for group g is defined as $\sigma_a = \sigma_t - \sum_{g'} \sigma_{g \to g'}$

When using the Los Alamos MENDF5 cross-section library with the codes there are numerous edit cross sections available for use in the Edit Module. Since these come from the MENDF file, they are called upon with special character names in the Edit Module as part of the EDXS input. These names are defined above.

REFERENCES

1. G. I. Bell and S. Glasstone, "Discrete Ordinates and Discrete S_n Methods," in Nuclear Reactor Theory, (Van Nostrand Reinhold, New York, 1970), Chap. 5, pp. 232-235.
2. B. G. Carlson and K. D. Lathrop, "Transport Theory-Method of Discrete Ordinates," in Computing Methods in Reactor Physics, H. Greenspan, C. N. Kelber and D. Okrent, Eds. (Gordon and Breach, New York, 1968), Chap. III, p. 185.
3. "DANTSYS: A Diffusion Accelerated Neutral Particle Code System," Los Alamos National Laboratory report LA-12969-M, Los Alamos National Laboratory (1995).
4. R. D. O'Dell, "Standard Interface Files and Procedures for Reactor Physics Codes, Version IV," Los Alamos Scientific Laboratory report LA-6941-MS (September 1977).
5. R. E. Alcouffe, "Diffusion Synthetic Acceleration Methods for the Diamond-Difference Discrete-Ordinates Equations," Nucl. Sci. Eng. 64, 344 (1977).
6. G. L. Ramone, M. L. Adams, and P. F. Novak, "A Transport Synthetic Acceleration Method for Transport Iterations," Nucl. Sci. Eng., 125, 257 (1997).
7. R. E. Alcouffe, "The Multigrid Method for Solving the Two-Dimensional Multigroup Diffusion Equation," Proc. Am. Nucl. Soc. Top. Meeting on Advances in Reactor Computations, Salt Lake City, Utah, March 28-31, 1983, Vol. 1, pp 340-351.
8. R. S. Baker, "A Stochastic First Collision Source for TWODANT," Proc. 8th Intl. Conf. on Radiation Shielding, Arlington, Texas, April 24-28, 1994, Vol. 1, pp. 157-164.
9. J. A. Dahl, B. D. Ganapol, and J. E. Morel, "Positive Scattering Cross Sections Using Constrained Least Squares," Proc. Joint Intl. Conf. on Mathematics and Computation, Reactor Physics, and Env. Analysis in Nucl. Applications, 1, 377, Madrid, Spain (1999).
10. S. A. Turner, "Automatic Mesh Coarsening for Discrete Ordinates Codes," Proc. Joint Intl. Conf. on Mathematics and Computation, Reactor Physics, and Env. Analysis in Nucl. Applications, 2, 1423, Madrid, Spain (1999).
11. R. S. Baker, "A Block Adaptive Mesh Refinement Algorithm for the Neutral Particle Transport Equation," Nucl. Sci. Eng., 141, 1 (2002).
12. R. E. Alcouffe and R. S. Baker, "Time-Dependent Deterministic Transport on Parallel Architectures Using PARTISN," Proc. 1998 ANS Radiation Protection and Shielding Topical Conf., 1, 335, Nashville, TN (1998).
13. R. S. Baker and R. E. Alcouffe, "Parallel 3-D SN Performance for DANTSYS/MPI on the Cray T3D," Proc. of the Joint Intl. Conf. on Mathematical Methods and Supercomputing for Nucl. Applications, 1, 377, Saratoga, NY (1997).
14. R. S. Baker and K. R. Koch, "An SN Algorithm for the Massively Parallel CM-200 Computer," Nucl. Sci. Eng., 128, 312 (1998).
15. R. S. Baker, "Parallel SN Methods for Orthogonal Grids," Proc. of the 9th SIAM Conf. on Parallel Processing, San Antonio, TX (1999).

16. R. D. O'Dell and R. E. Alcouffe, "Transport Calculations for Nuclear Analysis: Theory and Guidelines for Effective Use of Transport Codes," Los Alamos National Laboratory report LA-10983-MS (September 1987).

17. W. W. Engle, Jr., "A USER'S MANUAL FOR ANISN, A One Dimensional Discrete Ordinates Transport Code With Anisotropic Scattering," Union Carbide report K-1693 (March 1967).

18. R. E. Alcouffe, "An Adaptive Weighted Diamond Differencing Method for Three-Dimensional XYZ Geometry," *Trans Am Nuc Soc.* **68**, Part A, 206 (1993).

APPENDIX A: SAMPLE INPUT

This appendix presents the printed output from a sample problem. The sample problem is a standard k_{eff} calculation with all input by means of card-images.

Sample Problem 1: Standard k_{eff} Calculation

Sample Problem 1 is a standard k_{eff} calculation for a three-dimensional model fast breeder reactor. Four energy-group cross sections are used and the scattering is assumed isotropic. Only the Input and Solver modules are executed in this sample problem, and the Solver Module is run in parallel (two processors).

The first page of the PARTISN output (page 2-97) lists the entire card-image input "deck" supplied to the PARTISN code for this sample problem. The code provides this card-image input listing unless the third entry on card 1, the entry NOLIST, is set to unity by the user. Note that numerous "comment cards" have been used in the card-image input using the slash (/) as described on page 2-20.

On page 2-98 of the problem output is a descriptive summary of the Title Card Control Parameters and a printout of the two title cards provided. This is followed by the message KEY END BLOCK-I READ, which indicates that all Block-I input has been successfully read and is ready for processing. Next appears the Block-I input summary followed by messages that both the Block-II and Block-III input card-images were successfully read.

On page 2-98 of the output is a descriptive summary of the Block-III card-image input pertaining to cross sections. Included in this summary is a listing of the cross-section types from the card-image library that can be used for edit purposes. These edit cross sections are written to the SNXEDT group-ordered cross-section interface file for use by the Edit Module, if desired. (See the tables "Edit Cross-Section Types by Position and Name" on page 2-81 and "MENDF Library Edit Cross Sections" on page 2-88.) The card-image cross-section library, provided directly in the card-image input, is read, and the header cards that were included in the library are printed for the user. For this sample problem cross sections for five isotopes have been provided. Default names are used for each isotope. The scattering is specified to be isotropic, and this is indicated by the entries "p0" under the column labeled Order. (The label "Order" refers to the Legendre order of expansion for the scattering and, since it is isotropic, only the P_0 Legendre polynomial term appears.)

On page 2-99 of the output the user is provided with a listing of all Nuclide and Material Mixing instructions provided in Block-IV of the card-image input. Isotope one ("iso1") is used for material one, isotope two for material two, etc. Zone one is composed solely

of material one, zone two of material two, etc. Densities have already been factored into the cross sections, so no modifications are necessary. The subsequent message KEY START MIX CARD XS indicates that the PARTISN Input Module is to begin creating the working cross-section files MACRXS and SNXEDT and the standard interface files NDXSRF and ZNATDN as described in the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Code Structure" starting on page 13-1. The last three KEY END messages on the page indicate that the cross-section mixing and processing operation was completed, the Block-V Solver Module input was read (and the SOLINP interface file created), and all Input Module operations were completed.

On page 2-99 the printed output from by the Solver Module begins. There is first presented a summary of the input parameters related to, or required by the Solver Module as provided (or defaulted). Note that for the input parameters two columns are provided: one labeled RAW INPUT and one labeled AS DEFAULTED. These columns list the values of the input parameters that the Solver Module actually uses. For example, under the heading Required Input, the value for the parameter IBR is listed as 0. (In the actual card-image input, no entry for IBR has been provided.) The default value of IBR (0.0) is, accordingly, assumed by the Solver Module and this value is provided under the AS DEFAULTED column.

On page 2-100 through page 2-101 of the output are more of the Solver input parameters and also listed are the Block-I input parameters that are carried over for use by the Solver Module. Here, for example, is indicated that the problem is x-y-z geometry (IGEOM= 14), four energy groups (NGROUP= 4), and S_8 quadrature is to be used (ISN= 8), etc.

On page 2-101 of the output is provided a recap of the assignment of materials to zones in terms of the algorithm described on page 2-53 under the ASGMOD ARRAY description in Block-IV. Below this is a summary of the discrete-ordinates quadrature quantities used for the calculation. For this problem the values printed are built-in S_8 quadrature values. Following this on the next page is a map of the problem geometry showing the coarse-mesh boundary locations, the zone number assigned to each coarse-mesh interval, and other pertinent information.

Page 2-102 lists the material names of materials for which cross-section data exist on the MACRXS interface file being used by the Solver Module. Next is provided a listing of the ZONE macroscopic cross sections used by the Solver Module. This print is optional and is controlled by the XSECTP entry in the Block-V input. In this sample the full table of ZONE macroscopic cross sections has been requested by setting XSECTP= 2. The PRINCIPAL CROSS SECTIONS are defined as the ZONE macroscopic values of χ (fission fraction), $\nu\sigma_f$, σ_t, and σ_a .* The scattering matrix terms correspond to the coef-

ficients in a Legendre expansion of the term $\sigma_{s,g' \to g}(\underline{r}, \mu_0)$ in Eq. (2) on page 12-11. The value of the Legendre order for the term is provided under the column labeled ORDER in the printout. The actual scatter matrix terms for scatter from energy-group h to energy-group g are listed across the page in the sequence

$$\sigma_{s,h \to g}\ \sigma_{s,(h-1) \to g}\ \sigma_{s,(h-2) \to g}\ , \text{ etc.}$$

The entries in the column labeled FIRST GRP in the printout give the value of the energy-group h, namely the first group in the listing which scatters into group g. For downscatter only problems, the value of h is the same as the group number g. For upscatter problems the value of h will not be the same as the value of g. At the bottom of the output page is information related to mesh coarsening, which is not used in this problem.

The Solver Module memory requirements are output in the storage map on page 2-103. The maximum number of 8-byte words used per processor (PE) is listed, as well as the individual module component requirements. The diffusion 3d and block module component requirements are per spatial cell, while the remainder are per PE. Information on the parallel layout of the problem is also contained in this section including the number of PE's used, the spatial load balancing (ALB), and the parallel computational efficiency (PCE).[13-15]

On page 2-104 is provided a summary description of iteration control criteria followed by the iteration monitor print. These items are fully described on page 7-19. It is noted that for this type of problem, a k_{eff} calculation, the eigenvalue is the value of k_{eff}. For the sample problem, then, $k_{eff} = 0.97346344$ is provided in the monitor print for outer iteration 5 under the column labeled K-EFF EIGENVALUE.

Then page 2-105 provides a balance table print for each energy group and the sum of the groups. The group-dependent quantities are defined and computed as follows:

(1) SOURCE = total inhomogeneous source = QG_g

$$QG_g = \sum_{i=1} Q_i V_i + \sum_{\mu_m < 0} w_m |\mu_m| A_{IT+1/2} QR_m + \sum_{\mu_m > 0} w_m \mu_m A_{1/2} QL_m \ ,$$

* Note that in this discussion, as in the chapter "ONEDANT, TWODANT, TWOHEX, TWODANT/GQ, and THREEDANT — Methods Manual" starting on page 12-1, a lower case sigma is used to represent a macroscopic cross section.

where Q_i is the inhomogeneous distributed source, QL_m is the left boundary (surface) source, QR_m is the right boundary (surface) source, V_i is the "volume" of spatial mesh interval i, $A_{IT+1/2}$ is the surface area at the rightmost boundary of the system, and $A_{1/2}$ is the surface area at the leftmost boundary of the system (similar boundary source terms exist for the top, bottom, front, and back faces);

(2) FISSION SOURCE = total fission source to the group g = FG_g

For a k-$_{eff}$ eigenvalue problem,

$$FG_g = \frac{1}{k_{eff}} \sum_{h=1}^{NGROUP} \sum_{i=1}^{IT} \chi_{g,i} (\nu\sigma_f)_{h,i} \phi_{h,i} V_i ;$$

For a source with fission problem,

$$FG_g = \sum_{h=1}^{NGROUP} \sum_{i=1}^{IT} \chi_{g,i} (\nu\sigma_f)_{h,i} \phi_{h,i} V_i ;$$

(3) IN SCATTER = in scatter source to group g from other groups = SIN_g

$$SIN_g = \sum_{\substack{h=1 \\ h \neq g}}^{NGROUP} \sum_{i=1}^{IT} (\sigma_{s,h \to g})_i \phi_{h,i} V_i ;$$

(4) SELF SCATTER = self-scatter (within group scatter) in group g = SS_g

$$SS_g = \sum_{i=1}^{IT} (\sigma^o_{s,g \to g})_i \, \phi_{g,i} \, V_i ;$$

(5) OUT SCATTER = out-scatter from group g to all other groups = $SOUT_g$

$$SOUT_g = \sum_{i=1}^{IT} (\sigma'_{t,g})_i \, \phi_{g,i} \, V_i - ABG_g - SS_g ;$$

where $\sigma'_{t,g}$ is the total cross section for group g plus any buckling "absorption" plus any "time absorption" (α / v_g);

(6) ABSORPTION = absorption in group g = ABG_g

$$ABG_g = \sum_{i=1}^{IT} (\sigma'_{a,g})_i \, \phi_{g,i} \, V_i \; ;$$

where $\sigma'_{a,g}$ is the absorption cross section for group g plus any buckling "absorption"' plus any "time absorption" (α / v_g);

(7) RIGHT LEAKAGE = net flow out of system right boundary = RL_g (similar terms are used for the other faces)

$$RL_g = \sum_{\mu_m > 0} w_m \mu_m A_{IT+1/2} \psi_{m,IT+1/2} - \sum_{\mu_m < 0} w_m |\mu_m| A_{IT+1/2} \psi_{m,IT+1/2} \; ;$$

(8) NET LEAKAGE = net flow from system (all boundaries) = NL_g

(9) PARTICLE BALANCE = BAL_g

$$BAL_g = 1 - \frac{NL_g + ABG_g + SOUT_g}{QG_g + FG_g + SIN_g};$$

(10) NPROD = spectrum of neutrons causing fissions = $NPROD_g$

$$NPROD_g = \frac{1}{N} \sum_{i=1}^{IT} (\nu\sigma_f)_{g,i} \phi_{g,i} V_i \; ;$$

where - $N = \sum_{g=1}^{NGROUP} \sum_{i=1}^{IT} (\nu\sigma_f)_{g,i} \phi_{g,i} V_i$.

(11) TIME SOURCE = incoming source at time-edge (time-dependent calculations only)

Following this is the timing history of the run.

The final page (page 2-106), provides the RUN HIGHLIGHTS for the sample problem execution.

It is to be noted that no Edit Module output appears in the output of this sample problem. The reason for this is that there is no edit input available. No Edit Module input (Block-VI of the card-image input) is provided in the input "deck" and no EDITIT

binary interface file (containing previously created Edit Module input) was in existence at the time of the sample problem execution.

Sample Problem: Output Listing

```
1
 ********************************************************************************************************************************************
 ********************************************************************************************************************************************
 *
 *             generalized input module run on 08/22/02 with solver version 07-25-02    beta release 2.92     machine thepolic
 *
 ********************************************************************************************************************************************
 *                   ***************************************************************************************
 *                   *                                                                                         *
 *                   *                    ...listing of cards in the input stream...                           *
 *                   *                                                                                         *
 *                   *    1.     2          0                                                                  *
 *                   *    2.  3D model of a small FBR (Benchmark prob from Japan).                              *
 *                   *    3.  Cross sections from input...                                                      *
 *                   *    4. /                                                                                  *
 *                   *    5. / block 1                                                                          *
 *                   *    6. /                                                                                  *
 *                   *    7.   igeom=x-y-z ngroup=4 isn=8 iquad=1                                               *
 *                   *    8.   niso=5 mt=5 nzone=5                                                              *
 *                   *    9.   im=8 it=14 jm=8 jt=14 km=4 kt=30                                                 *
 *                   *   10.   maxscm=300000 maxlcm=600000                                                      *
 *                   *   11.   T                                                                               *
 *                   *   12. /                                                                                 *
 *                   *   13. / block 2                                                                         *
 *                   *   14. /                                                                                 *
 *                   *   15.  xmesh=0.0 15.0 30.0 35.0 40.0 45.0 50.0 55.0 70.0                                *
 *                   *   16.  xints=3 3 1 1 1 1 1 3                                                            *
 *                   *   17.  ymesh=0.0 5.0 15.0 30.0 40.0 45.0 50.0 55.0 70.0                                 *
 *                   *   18.  yints=1 2 3 2 1 1 1 3                                                            *
 *                   *   19.  zmesh=0.0 20.0 75.0 130.0 150.0                                                  *
 *                   *   20.  zints=4 11 11 4                                                                  *
 *                   *   21.  zones=3r2 2r3 2r2 5; 7r2 5; 6r2 2r5; 5r2 3r5; 4r2 4r5; 2r2 6r5;                  *
 *                   *   22.                2 7r5; 8r5;                                                        *
 *                   *   23.         3r1 2r3 2r1 5; 7r1 5; 6r1 2r5; 5r1 3r5; 4r1 4r5; 2r1 6r5;                 *
 *                   *   24.                1 7r5; 8r5;                                                        *
 *                   *   25.         3r1 2r3 2r1 5; 7r1 5; 6r1 2r5; 5r1 3r5; 4r1 4r5; 2r1 6r5;                 *
 *                   *   26.                1 7r5; 8r5;                                                        *
 *                   *   27.         3r2 2r3 2r2 5; 7r2 5; 6r2 2r5; 5r2 3r5; 4r2 4r5; 2r2 6r5;                 *
 *                   *   28.                2 7r5; 8r5;                                                        *
 *                   *   29.   T                                                                               *
 *                   *   30. /                                                                                 *
 *                   *   31. / block 3                                                                         *
 *                   *   32. /                                                                                 *
 *                   *   33.  lib=odnimp                                                                       *
 *                   *   34.  ihm=8 iht=4 ihs=5           / 1 activity position in slot 1                      *
 *                   *   35.  ititl=1 ifido=2                                                                  *
 *                   *   36. /                                                                                 *
 *                   *   37. / inline cross sections in free-form FIDO format follow this block                *
 *                   *   38. /                                                                                 *
 *                   *   39.   T                                                                               *
 *                   *   40. core                                                                              *
 *                   *   41. 1. 0.00745551 0.0206063 0.114568 0.0704326 3r0.0 /                                *
 *                   *   42. 1. 0.0035254 0.00610571 0.205177 0.195443 0.0347967 2r0.0 /                       *
 *                   *   43. 1. 0.00780136 0.00691403 0.329381 0.320586 0.00620863 0.00188282 0.0 /            *
 *                   *   44. 1. 0.0274496 0.0260689 0.38981 0.36236 9.92975e-4 7.07208e-7 0.0 /                *
 *                   *   45.   T                                                                               *
 *                   *   46. axblkt                                                                            *
 *                   *   47. 1. 0.00535418 0.013177 0.116493 0.0716044 3r0.0 /                                 *
 *                   *   48. 1. 0.00148604 1.26026e-4 0.220521 0.210436 0.037317 0.0 0.0 /                     *
 *                   *   49. 1. 0.005353 1.5238e-4 0.344544 0.337506 0.00859855 0.00221707 0.0 /               *
 *                   *   50. 1. 0.0134694 7.87302e-4 0.388356 0.374886 0.0016853 6.68299e-7 0.0 /              *
 *                   *   51.   T                                                                               *
 *                   *   52. nacrod                                                                            *
 *                   *   53. 1.  3.10744e-4 0.0 0.0658979 0.0474407 3r0.0 /                                    *
 *                   *   54. 1.  1.13062e-4 0.0 0.10981 0.106142 0.0176894 0.0 0.0 /                           *
 *                   *   55. 1.  4.48988e-4 0.0 0.186765 0.185304 0.00355466 4.57012e-4 0.0 /                  *
 *                   *   56. 1.  0.00107518 0.0 0.209933 0.208858 0.0010128 1.77599e-7 0.0 /                   *
 *                   *   57.   T                                                                               *
 *                   *   58. cntrod                                                                            *
 *                   *   59. 1. 0.00597638 0.0 0.184333 0.134373 3r0.0 /                                       *
 *                   *   60. 1. 0.0176941 0.0 0.366121 0.318582 0.0437775 0.0 0.0 /                            *
 *                   *   61. 1. 0.0882741 0.0 0.615527 0.519591 0.0298432 2.06054e-4 0.0 /                     *
 *                   *   62. 1. 0.476591 0.0 1.09486 0.618265 0.00766209 8.71188e-7 0.0 /                      *
 *                   *   63.   T                                                                               *
 *                   *   64. radblk                                                                            *
 *                   *   65. 1. 0.00743283 0.0189496 0.119648 0.0691158 3r0.0 /                                *
 *                   *   66. 1. 0.0019906  1.75265e-4 0.242195 0.230626 0.0404132 0.0 0.0 /                    *
 *                   *   67. 1. 0.00679036 2.06978e-4 0.356476 0.348414 0.00957027 0.00268621 0.0 /            *
 *                   *   68. 1. 0.0158015 0.00113451 0.379433 0.363631 0.00127195 1.99571e-7 0.0 /             *
 *                   *   69.   T                                                                               *
 *                   *   70. /                                                                                 *
 *                   *   71. / block 4                                                                         *
 *                   *   72. /                                                                                 *
 *                   *   73.  matls=isos                                                                       *
 *                   *   74.  assign=matls                                                                     *
 *                   *   75.   T                                                                               *
 *                   *   76. /                                                                                 *
 *                   *   77. / block 5                                                                         *
 *                   *   78. /                                                                                 *
 *                   *   79.  ievt=1 ibl=1 ibb=1 ibfrnt=0 norm=1 iitl=1 oitm=11 xsectp=2                       *
 *                   *   80.  epsi=1.0e-3 chi=0.583319 0.40545 0.011231 0.0                                    *
 *                   *   81.  kcalc=1 nchunk=7                                                                 *
 *                   *   82.  T                                                                                *
 *                   *                                                                                         *
 *                   ***************************************************************************************
1
 ********************************************************************************************************************************************
 ********************************************************************************************************************************************
 *
 *                                                  case title
 *
 ********************************************************************************************************************************************
 ************************
 *key start case input  *
 ************************
```

```
 *                                   2  nhead   number of title cards to follow
 *                                   0  notty   0/1  no/yes  suppress on-line terminal output
 *                                   0  nolist  0/1  no/yes  suppress input listing
 *                                   0  restart 0/1 no/yes read td_restart file
 *
 *                        ************************************************************************
 *                        *                                                                      *
 *                        *  3D model of a small FBR (Benchmark prob from Japan).                *
 *                        *  Cross sections from input...                                        *
 *                        *                                                                      *
 *                        ************************************************************************
 ************************
 *key end   block i read*
 ************************
 ********************************************************************************************************************************************************
1
 ********************************************************************************************************************************************************
 ********************************************************************************************************************************************************
 *
 *                                        ...block i - controls and dimensions...
 *
 ********************************************************************************************************************************************************
 *
 *
 *                                             ...dimensions ...
 *
 *                        14  igeom   14/15  x-y-z/r-z-theta
 *                         4  ngroup  number of energy groups
 *                         8  isn     angular quadrature order
 *                         1  iquad   1/2/3/4/5/6/7/8/9 - source of quadrature constants (default=1)
 *                                    1 traditional built-in Pn set (1-D) or old twotran built-in set
 *                                    2 traditional built-in DPn constants (1-D), tri DPn-Tchev product (2/3-D)
 *                                    3 card input
 *                                    4 galerkin (based on triangular hybrid product set)
 *                                    5 hybrid product set (triangular arrangement)
 *                                    6 product set (rectangular arrangement)
 *                                    7 sncon file
 *                                    8 DPn-Tchebychev (sq product set)
 *                                    9 Generalized Quadrature
 *                         5  niso    number of input isotopes (from isotxs, grupxs, or cards)
 *                         5  mt      number of permanent materials
 *                         5  nzone   number of zones
 *                         8  im      number of coarse mesh x intervals
 *                        14  it      number of fine mesh intervals
 *                         8  jm      number of coarse mesh y intervals
 *                        14  jt      number of fine mesh y intervals
 *                         4  km      number of coarse mesh z intervals
 *                        30  kt      number of fine mesh z intervals
 *
 *                                               ...storage...
 *
 *                                              maxlcm=  600000
 *                                              maxscm=  300000
 *
 ********************************************************************************************************************************************************
 ******************************
 *key end   block ii read-geom*
 ******************************
 *
 *                       14  igeom    1/2/3/6/7/11/14/15
 ******************************
 *key end   block iii read-xs *
 ******************************
1
 ********************************************************************************************************************************************************
 ********************************************************************************************************************************************************
 *
 *                                        ...block iii - cross section library...
 *
 ********************************************************************************************************************************************************
 *
 *                                              ...library source...
 *                                                   lib=odninp
 *
 *                                           ...card library parameters (array name = cards)...
 *
 *
 *                        0  maxord  maximum legendre order to be found in input cross sections
 *                        8  ihm     last position in cross section table
 *                        4  iht     position of total cross section
 *                        5  ihs     position of self scatter cross section
 *                        2  ifido   -1/0/1/2 - dtf(4e18.0)/dtf/fixed fido/free fido library
 *                        1  ititl   0/1 - no/yes there is a title card before each table
 *                        0  i2lp1   0/1 - no/yes library higher order scattering contains 2l+1 factor
 *                        0  savbxs  0/1 - no/yes save binary xslib file (filename=bxslib)
 *                        0  kwikrd  0/1 - full fido read/quick fido read (default=quick)
 *
 *
 *                                              ...energy structure...
 *
 *   group      chi          vel       lower bound   upper bound   group      chi          vel       lower bound   upper bound
 *   -----  -----------  -----------  -----------   -----------   -----  -----------  -----------  -----------   -----------
 *       1  0.00000E+00  0.00000E+00  0.00000E+00   0.00000E+00       3  0.00000E+00  0.00000E+00  0.00000E+00   0.00000E+00
 *       2  0.00000E+00  0.00000E+00  0.00000E+00   0.00000E+00       4  0.00000E+00  0.00000E+00  0.00000E+00   0.00000E+00
 *
 *                                         last neutron group(lng) is number   4
 *
 *
 *                        0  balxs   -1/0/1 - adjust absorption/no/adjust self scatter to force xs balance
 *
 *
 *                                              ...edit position names...
 *
 *                                                                 card
 *                                             position edname position
 *                                             -------- ------ --------
 *                                                 1    chi
 *                                                 2    nusigf     3
 *                                                 3    total      4
```

```
*                                              4   abs        2
*                                              5   edit1      1
******************************
*key start card libe read    *
******************************
*
*                                    ...header cards from the card library...
*
*                           isotope isotope
*                           number   name   order          header card
*                           ------  ------- -----    ------------------------
*                                1.  iso1    p0    -  core
*                                2.  iso2    p0    -  axblkt
*                                3.  iso3    p0    -  nacrod
*                                4.  iso4    p0    -  cntrod
*                                5.  iso5    p0    -  radblk
******************************
*key end   card libe read    *
******************************
******************************
*key end   block iv read-mats*
******************************
*
*
*
*********************************************************************************************************************************************
*********************************************************************************************************************************************


1********************************************************************************************************************************************
 *********************************************************************************************************************************************
 *
 *                                    ... mixing instructions ...
 *********************************************************************************************************************************************
 *
 *          mix      comp       density    comp       density etc.
 *          ---      ----       -------    ----       -------
 *
 *     matls
 *     ------
 *
 *       1. iso1     iso1     1.00000E+00,
 *       2. iso2     iso2     1.00000E+00,
 *       3. iso3     iso3     1.00000E+00,
 *       4. iso4     iso4     1.00000E+00,
 *       5. iso5     iso5     1.00000E+00,
 ************************
 *key start mix card xs *
 ************************
 ************************
 *key end   mix card xs *
 ************************
 *********************************************************************************************************************************************
 ******************************
 *key end   block v read-solvr*
 ******************************
 ************************
 *key end   input module*
 ************************
 *
 *
 *********************************************************************************************************************************************
 *********************************************************************************************************************************************
1                    this partisn problem run on 08/22/02 with solver version 08-14-02    beta release 2.92     machine thepolic
 * 3D model of a small FBR (Benchmark prob from Japan).
 * Cross sections from input...
 *
 *********************************************************************************************************************************************
 *
 *                                    ...block v -- solver input...
 *
 *********************************************************************************************************************************************
 *           raw     as
 *          input  defaulted
 *          -----  --------
 *
 *                                    ...required input(array name = solin)...
 *
 *          1         1      ievt   0/1/2/3/4 - type of calculation
 *                                      0 inhomogeneous source
 *                                      1 k-effective
 *                                      2 alpha or time absorption search
 *                                      3 concentration search
 *                                      4 delta(i.e. dimension) search
 *          0         0      isct   legendre order of scattering
 *          0         0      ith    0/1 - direct/adjoint - mode of calculation (default=direct)
 *          1         1      ibl    0/1 - left boundary condition
 *                                     vacuum/reflective
 *          0         0      ibr    0/1/3 - right boundary condition
 *                                     vacuum/reflective/white
 *          0         0      ibt    0/1/2/3 - top boundary condition
 *                                     vacuum/reflective/periodic/white
 *          1         1      ibb    0/1/2 - bottom boundary condition
 *                                     vacuum/reflective/periodic
 *          0         0      ibbk   0/1/2/3 - back boundary condition
 *                                     vacuum/reflective/periodic/white
 *          0         0      ibfr   0/1/2 - front boundary condition
 *                                     vacuum/reflective/periodic
 *
 *
 *                                    ...convergence controls(array name = iter)...
 *
 *       1.000E-03 1.000E-03  epsi   inner iteration convergence criterion (default=0.0001)
 *           1         1      iitl   maximum number of inner iterations per group until fission source is near
 *                                     convergence, i.e. lambda is near convergence.(default=1)
 *          30        30      iitm   maximum number of inner iterations per group when close to fission source convergence
 *                                     (default calculated)
 *          11        11      oitm   maximum number of outer iterations (default=20)
 *
```

```
 *
 *
 ********************************************************************************************************************************************
 ********************************************************************************************************************************************
1********************************************************************************************************************************************
 ********************************************************************************************************************************************
 *
 *                                ...block v -- solver input (continued)...
 *
 ********************************************************************************************************************************************
 *           raw     as
 *         input  defaulted
 *         -----  --------
 *
 *
 *                                      ...miscellaneous parameters(array name = misc)...
 *
 *      0.000E+00 0.000E+00  bhgt   buckling height
 *      0.000E+00 0.000E+00  bwth   buckling width
 *      1.000E+00 1.000E+00  norm   normalization factor
 *
 *          0         0      nosigf set fissions zero when source calculation (0/1 no/yes)
 *          0         0      influx 0/1 no/yes - read input flux from file rtflux (atflux for adjoint)
 *          0         0      insors 0/1 no/yes - read input source from file fixsrc
 *          0         0      nodal  0/1/2 diamond/exp disc/linear disc spatial differencing
 *          0         0      iufcs  fcsrc option (0/1/2/3/4 no/point ana/ray tracing/point beam/umcflux)
 *      10000     10000      fcnray number of rays/trial for fcsrc rt option
 *         25        25      fcntr  number of trials for fcsrc rt option
 *          0         0      fcrstrt 0/1/2 - no/yes/gen restart from ucflux file
 *      1.000E-06 1.000E-06  fcwco  weight cutoff for fcsrc rt option
 *          1         1      trnsol 0/1/2/3 - vect diag line/seq diag plane/block jacobi/block sweep transport solver
 *          0         0      srcacc 0/1/2 - DSA/TSA/none - source iteration acceleration option
 *          0         0     diffsol 0/1/2/3/4/5/6/7 - multigrid/CG 3 line/CG 1 line/red-black line/ MG 1 line/ CG-MG1L/MG Pln-Z/CG-
RBPT
 *          0         0      wgtdia 0/1 - no/yes weighted diamond in angle
 *      0.000E+00 0.000E+00  tsabeta value of beta for TSA acceleration
 *      1.000E-02 1.000E-02  tsaepsi convergence criteria for TSA sweeps
 *          0         0      tsaits max number of TSA iterations
 *          2         2      tsasn  Sn order for TSA sweeps
 *          1         1      npey   number of PE's along the J axis
 *          2         2      npez   number of PE's along the K axis
 *          7         7      nchunk number of I spatial planes solved per communications step
 *          0         0      angflx 0/1 - no/yes angular flux on disk
 *          0         0      flxmom 0/1 - no/yes flux moments on disk
 *          0         0      lssn   0/1 - no/yes least squares adjustment of cross section moments
 *          0         0      lsxs   0/1/2 none/write/read least squares adjusted moments to/from ascii file lsxsdat
 *
 *
 *                                        ...output controls(array name = solout)...
 *
 *                        0  fluxp  0/1/2   none/isotropic/all moments - flux print
 *                        2  xsectp 0/1/2   none/principal/all - macroscopic cross section print
 *                        0  fissrp 0/1     no/yes - print final fission source rate
 *                        0  sourcp 0/1/2/3 no/as read/normalized/both - print inhomogeneous source
 *                        0  angp   0/1     no/yes - print angular fluxes
 *                        0  raflux 0/1     no/yes - write angular fluxes to file raflxm or aaflxm(if ith=1)
 *                        0  rmflux 0/1     no/yes - write flux moments to file rmflux
 *                        0  iavtar 0/1     no/yes - write xmfluxa-asc for avatar
 *                        0  balp   0/1     no/yes - print coarse mesh balances
 *                        0  asleft 0/i     no/yes - write right-going angular edge flux at plane i
 *                        0  asrite 0/i     no/yes - write left-going angular edge flux at plane i
 *                        0  asbott 0/j     no/yes - write top-going angular edge flux at plane j
 *                        0  astop  0/j     no/yes - write bottom-going angular edge flux at plane j
 *                        0  asfrnt 0/k     no/yes - write back-going angular edge flux at plane k
 *                        0  asback 0/k     no/yes - write front-going angular edge flux at plane k
 *
 *
 *
 ********************************************************************************************************************************************
 ********************************************************************************************************************************************
1********************************************************************************************************************************************
 ********************************************************************************************************************************************
 *
 *                                ...block v -- solver input (continued)...
 *
 ********************************************************************************************************************************************
 *
 *
 *
 *                                     ...parameters inferred from input arrays...
 *
 *                                      1  inchi  0/1/2 none/one chi/zonewise chi
 *                                      0  isdenx 0/1/n - none/x density vector/full matrix
 *                                      0  isdeny 0/1 no/yes - use y density vector
 *                                      0  iqan   source anisotropy
 *                                      0  isorse number of source moments input
 *                                      0  isorsx number of sourcx moments input
 *                                      0  isorsy number of sourcy moments input
 *                                      0  isorsz number of sourcz moments input
 *                                      0  isorsf number of sourcf moments input
 *                                         boundary source parameters
 *                                         -1 isotropic
 *                                          0 none
 *                                          1 full angular
 *                                          2 space energy angle vectors
 *                                          3 group angle vectors & space array
 *                                          4 group vector space-angle array
 *                                          5 group vector angle-space array
 *                                          6 interface file
 *                                      0  iql  left boundary source
 *                                      0  iqr  right boundary source
 *                                      0  iqt  top boundary source
 *                                      0  iqb  bottom boundary source
 *                                      0  iqf  front boundary source
 *                                      0  iqk  back boundary source
 *
 *
 *
 ********************************************************************************************************************************************
```

```
 ****************************************************************************************************************************************************
1****************************************************************************************************************************************************
 ****************************************************************************************************************************************************
 *
 *
 *                                        ...parameters from block i...
 *
 *                          14  igeom   14/15  x-y-z/r-z-theta
 *                           4  ngroup  number of energy groups
 *                           8  isn     angular quadrature order
 *                           1  iquad   1/2/3/4/5/6/7/8/9 - source of quadrature constants (default=1)
 *                                      1 traditional built-in Pn set (1-D) or old twotran built-in set
 *                                      2 traditional built-in DPn constants (1-D), tri DPn-Tchev product (2/3-D)
 *                                      3 card input
 *                                      4 galerkin (based on triangular hybrid product set)
 *                                      5 hybrid product set (triangular arrangement)
 *                                      6 product set (rectangular arrangement)
 *                                      7 sncon file
 *                                      8 DPn-Tchebychev (sq product set)
 *                                      9 Generalized Quadrature
 *                           5  mt      number of permanent materials
 *                           5  nzone   number of zones
 *                           8  im      number of coarse mesh x intervals
 *                           8  jm      number of coarse mesh y intervals
 *                           4  km      number of coarse mesh z intervals
 *                          14  it      number of fine mesh x intervals
 *                          14  jt      number of fine mesh y intervals
 *                          30  kt      number of fine mesh z intervals
 *
 *
 *
 *
 ****************************************************************************************************************************************************
 ****************************************************************************************************************************************************
 *
 *
 *                                     ...material assignments to zones...
 *
 *
 ******************************
 *key start matls to zones    *
 ******************************
 *
 *                                              zone     material
 *                                            no. name   no. name
 *
 *                                             1 zone1    1 iso1
 *                                             2 zone2    2 iso2
 *                                             3 zone3    3 iso3
 *                                             4 zone4    4 iso4
 *                                             5 zone5    5 iso5
 ****************************************************************************************************************************************************
 ****************************************************************************************************************************************************
 *
 * zero flux
 *
 *
 ****************************************************************************************************************************************************
 ****************************************************************************************************************************************************
1****************************************************************************************************************************************************
 *
 ************************
 *key start sn constants*
 ************************
 *
 * s 8 constants for groups  1 to  4
 *                 mu            eta           xi            weight
 *
 *     1      0.19232747E+00  0.96229948E+00  0.19232747E+00  0.14598559E-01
 *     2      0.57735027E+00  0.79352178E+00  0.19232746E+00  0.11656904E-01
 *     3      0.19232747E+00  0.79352178E+00  0.57735027E+00  0.11656904E-01
 *     4      0.79352178E+00  0.57735027E+00  0.19232746E+00  0.11656904E-01
 *     5      0.57735027E+00  0.57735027E+00  0.57735027E+00  0.11262900E-01
 *     6      0.19232747E+00  0.57735027E+00  0.79352178E+00  0.11656904E-01
 *     7      0.96229948E+00  0.19232747E+00  0.19232747E+00  0.14598559E-01
 *     8      0.79352178E+00  0.19232747E+00  0.57735027E+00  0.11656904E-01
 *     9      0.57735027E+00  0.19232747E+00  0.79352178E+00  0.11656904E-01
 *    10      0.19232747E+00  0.19232747E+00  0.96229948E+00  0.14598559E-01
1*************************************************material map*************************************************************************************
 ************************
 *key start material map*
 ************************
 * material map for k mesh intervals   1, to   4 and j mesh intervals  14, to   1, and i mesh intervals   1, to  14.
 *
 * 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 * 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 * 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 * 2 2 2 5 5 5 5 5 5 5 5 5 5 5
 * 2 2 2 2 2 2 5 5 5 5 5 5 5 5
 * 2 2 2 2 2 2 2 2 5 5 5 5 5 5
 * 2 2 2 2 2 2 2 2 2 5 5 5 5 5
 * 2 2 2 2 2 2 2 2 2 5 5 5 5 5
 * 2 2 2 2 2 2 2 2 2 2 5 5 5 5
 * 2 2 2 2 2 2 2 2 2 2 5 5 5 5
 * 2 2 2 2 2 2 2 2 2 2 5 5 5 5
 * 2 2 2 2 2 2 2 2 2 2 2 5 5 5
 * 2 2 2 2 2 2 2 2 2 2 2 5 5 5
 * 2 2 2 2 2 2 2 3 3 2 2 5 5 5
 * material map for k mesh intervals   5, to  15 and j mesh intervals  14, to   1, and i mesh intervals   1, to  14.
 *
 * 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 * 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 * 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 * 1 1 1 5 5 5 5 5 5 5 5 5 5 5
 * 1 1 1 1 1 1 5 5 5 5 5 5 5 5
 * 1 1 1 1 1 1 1 1 5 5 5 5 5 5
 * 1 1 1 1 1 1 1 1 1 5 5 5 5 5
 * 1 1 1 1 1 1 1 1 1 5 5 5 5 5
 * 1 1 1 1 1 1 1 1 1 1 5 5 5 5
```

```
 * 1 1 1 1 1 1 1 1 1 1 5 5 5 5
 * 1 1 1 1 1 1 1 1 1 1 5 5 5 5
 * 1 1 1 1 1 1 1 1 1 1 5 5 5 5
 * 1 1 1 1 1 1 1 1 1 1 5 5 5 5
 * 1 1 1 1 1 1 1 3 3 1 1 5 5 5
 * material map for k mesh intervals  16, to  26 and j mesh intervals  14, to   1, and i mesh intervals   1, to  14.
 *
 * 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 * 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 * 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 * 1 1 1 5 5 5 5 5 5 5 5 5 5 5
 * 1 1 1 1 1 1 5 5 5 5 5 5 5 5
 * 1 1 1 1 1 1 1 1 5 5 5 5 5 5
 * 1 1 1 1 1 1 1 1 1 1 5 5 5 5
 * 1 1 1 1 1 1 1 1 1 1 5 5 5 5
 * 1 1 1 1 1 1 1 1 1 1 5 5 5 5
 * 1 1 1 1 1 1 1 1 1 1 5 5 5 5
 * 1 1 1 1 1 1 1 1 1 1 5 5 5 5
 * 1 1 1 1 1 1 1 1 1 1 5 5 5 5
 * 1 1 1 1 1 1 1 3 3 1 1 5 5 5
1* material map for k mesh intervals  27, to  30 and j mesh intervals  14, to   1, and i mesh intervals   1, to  14.
 *
 * 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 * 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 * 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 * 2 2 2 5 5 5 5 5 5 5 5 5 5 5
 * 2 2 2 2 2 2 5 5 5 5 5 5 5 5
 * 2 2 2 2 2 2 2 2 5 5 5 5 5 5
 * 2 2 2 2 2 2 2 2 2 2 5 5 5 5
 * 2 2 2 2 2 2 2 2 2 2 5 5 5 5
 * 2 2 2 2 2 2 2 2 2 2 5 5 5 5
 * 2 2 2 2 2 2 2 2 2 2 5 5 5 5
 * 2 2 2 2 2 2 2 2 2 2 5 5 5 5
 * 2 2 2 2 2 2 2 2 2 2 5 5 5 5
 * 2 2 2 2 2 2 2 2 2 2 5 5 5 5
 * 2 2 2 2 2 2 2 3 3 2 2 5 5 5
 *
 ********************************************************************************************************************************
*********************************************************************************************************************************
1********************************************************************************************************************************
 *
 *                                        Three-dimensional coarse mesh geometry edit...
 *
 ********************************************************************************************************************************
 *
 *
 ******************************
 *key start geometry edit     *
 ******************************
 *   n    xmesh       xint   deltax         ymesh       yint   deltay         zmesh       zint   deltaz
 *
 *   0  0.000E+00                         0.000E+00                         0.000E+00
 *   1  1.500E+01       3   5.000E+00      5.000E+00       1   5.000E+00      2.000E+01       4   5.000E+00
 *   2  3.000E+01       3   5.000E+00      1.500E+01       2   5.000E+00      7.500E+01      11   5.000E+00
 *   3  3.500E+01       1   5.000E+00      3.000E+01       3   5.000E+00      1.300E+02      11   5.000E+00
 *   4  4.000E+01       1   5.000E+00      4.000E+01       2   5.000E+00      1.500E+02       4   5.000E+00
 *   5  4.500E+01       1   5.000E+00      4.500E+01       1   5.000E+00
 *   6  5.000E+01       1   5.000E+00      5.000E+01       1   5.000E+00
 *   7  5.500E+01       1   5.000E+00      5.500E+01       1   5.000E+00
 *   8  7.000E+01       3   5.000E+00      7.000E+01       3   5.000E+00
 *
 ********************************************************************************************************************************
1
 ********************************************************************************************************************************
 ********************************************************************************************************************************
 *
 *                    ...cross section related data from file macrxs  134518082220 version  1 ...
 *
 ********************************************************************************************************************************
 *
 *
 *  1 iso1      2 iso2      3 iso3      4 iso4      5 iso5
 *
 *
 *
 ********************************************************************************************************************************
1
 ********************************************************************************************************************************
 ********************************************************************************************************************************
 *
 *                                       ...cross sections for legendre orders up to p 0...
 *
 ********************************************************************************************************************************
 *
 *
 ******************************
 *key start mac cross sections*
 ******************************
 *
 ********************************************************* g r o u p   1 ********************************************************
 *
 *                                         ...principal cross sections...
 *
 *                         zone          chi         nu*fission      total       absorption
 *                       no.  name
 *                         1  zone1     5.8332E-01    2.0606E-02    1.1457E-01    7.4555E-03
 *                         2  zone2     5.8332E-01    1.3177E-02    1.1649E-01    5.3542E-03
 *                         3  zone3     5.8332E-01    0.0000E+00    6.5898E-02    3.1074E-04
 *                         4  zone4     5.8332E-01    0.0000E+00    1.8433E-01    5.9764E-03
 *                         5  zone5     5.8332E-01    1.8950E-02    1.1965E-01    7.4328E-03
 *
 *                                              ...scattering matrices...
 *                                                 (2l+1 not included)
 *
 *   zone   order  first grp    cross sections
 *     1      0      1          7.0433E-02
 *     2      0      1          7.1604E-02
 *     3      0      1          4.7441E-02
```

```
*    4      0      1         1.3437E-01
*    5      0      1         6.9116E-02
*
************************************************** g r o u p   2 ***************************************************
*
*                                        ...principal cross sections...
*
*                        zone          chi         nu*fission      total       absorption
*                      no.  name
*                        1  zone1     4.0545E-01    6.1057E-03    2.0518E-01    3.5254E-03
*                        2  zone2     4.0545E-01    1.2603E-04    2.2052E-01    1.4860E-03
*                        3  zone3     4.0545E-01    0.0000E+00    1.0981E-01    1.1306E-04
*                        4  zone4     4.0545E-01    0.0000E+00    3.6612E-01    1.7694E-02
*                        5  zone5     4.0545E-01    1.7527E-04    2.4220E-01    1.9906E-03
*
*                                              ...scattering matrices...
*                                                 (2l+1 not included)
*
*   zone   order  first grp    cross sections
*     1      0      2         1.9544E-01    3.4797E-02
*     2      0      2         2.1044E-01    3.7317E-02
*     3      0      2         1.0614E-01    1.7689E-02
*     4      0      2         3.1858E-01    4.3778E-02
*     5      0      2         2.3063E-01    4.0413E-02
*
************************************************** g r o u p   3 ***************************************************
*
*                                        ...principal cross sections...
*
*                        zone          chi         nu*fission      total       absorption
*                      no.  name
*                        1  zone1     1.1231E-02    6.9140E-03    3.2938E-01    7.8014E-03
*                        2  zone2     1.1231E-02    1.5238E-04    3.4454E-01    5.3530E-03
*                        3  zone3     1.1231E-02    0.0000E+00    1.8677E-01    4.4899E-04
*                        4  zone4     1.1231E-02    0.0000E+00    6.1553E-01    8.8274E-02
*                        5  zone5     1.1231E-02    2.0698E-04    3.5648E-01    6.7904E-03
*
*                                              ...scattering matrices...
*                                                 (2l+1 not included)
*
*   zone   order  first grp    cross sections
*     1      0      3         3.2059E-01    6.2086E-03    1.8828E-03
*     2      0      3         3.3751E-01    8.5985E-03    2.2171E-03
*     3      0      3         1.8530E-01    3.5547E-03    4.5701E-04
*     4      0      3         5.1959E-01    2.9843E-02    2.0605E-04
*     5      0      3         3.4841E-01    9.5703E-03    2.6862E-03
*
************************************************** g r o u p   4 ***************************************************
*
*                                        ...principal cross sections...
*
*                        zone          chi         nu*fission      total       absorption
*                      no.  name
*                        1  zone1     0.0000E+00    2.6069E-02    3.8981E-01    2.7450E-02
*                        2  zone2     0.0000E+00    7.8730E-04    3.8836E-01    1.3469E-02
*                        3  zone3     0.0000E+00    0.0000E+00    2.0993E-01    1.0752E-03
*                        4  zone4     0.0000E+00    0.0000E+00    1.0949E+00    4.7659E-01
*                        5  zone5     0.0000E+00    1.1345E-03    3.7943E-01    1.5802E-02
*
*                                              ...scattering matrices...
*                                                 (2l+1 not included)
*
*   zone   order  first grp    cross sections
*     1      0      4         3.6236E-01    9.9298E-04    7.0721E-07
*     2      0      4         3.7489E-01    1.6853E-03    6.6830E-07
*     3      0      4         2.0886E-01    1.0128E-03    1.7760E-07
*     4      0      4         6.1827E-01    7.6621E-03    8.7119E-07
*     5      0      4         3.6363E-01    1.2719E-03    1.9957E-07
*
*
************************************************************************************************************************
************************************************************************************************************************
1***********************************************************************************************************************
 *
 ************************************
 *key start mesh potential info     *
 ************************************
 *
 * weight factor a:                  9.00000E-01
 * weight factor b:                  1.00000E-01
 *
 * energy weighting function:        2.50000E-01
 *                                   2.50000E-01
 *                                   2.50000E-01
 *                                   2.50000E-01
 *
 * user input min potential:         1.25000E-01
 * user input max source slope:      5.00000E-01
 *
 * initial max aspect ratio:         1.00000E+00
 *
 ************************************************************************************************************************
1***********************************************************************************************************************
 *
 ************************
 *key start storage map *
 ************************
 *
 *  partisn storage summary...
 *
 *     max words/PE allocated in alloc3d for this problem     237683
 *
 *     words/PE for the geom3d module              322
 *     words/PE for the smpn3d module              198
 *     words/PE for the fcsrc3d module               0
 *     words/PE for the bal3d module               110
 *     words/PE for the transport module            40
 *     words/PE for the xsec module                217
 *     words/PE for the time module                  0
```

```
 *     max words/PE for the meshpot3d module       340
 *     words/cell for the diffusion3d module     41.2
 *     words/cell for the block module           34.0
 *
 *     max cells/PE      2940
 *     min cells/PE      2940
 *     total cells       5880
 *
 * number of PE's used =    2
 * PCE = 0.982,  SN =   8
 * ALB = 1.000
 *
 ********************************************************************************************************************************
 ********************************************************************************************************************************
1
 ********************************************************************************************************************************
 ********************************************************************************************************************************
 *
 *                                ...iteration controls and criteria...
 *
 ********************************************************************************************************************************
 *
 *
 *                                        ***iteration criteria***
 *
 *                                           transport inners
 *                                           ----------------
 *         criterion    quantity to test                          value     action taken if value exceeded
 *         ---------    ----------------                          -----     ------------------------------
 *            iitl  - inner iteration count until near lambda       1         terminates inners
 *                    (i.e. fission source) convergence
 *            iitm  - inner iteration count when near lambda       30         terminates inners
 *                    (i.e. fission source) convergence
 *            epsi  - fractional ptwise flux change           1.00E-03       does another inner
 *                    per inner
 *
 *                                         diffusion sub-outers
 *                                         --------------------
 *         criterion    quantity to test                          value     action taken if value exceeded
 *         ---------    ----------------                          -----     ------------------------------
 *            oitmd - sub-outer iteration count                    22         terminates sub-outers
 *            eps   - diffusion lambda-1.0 (see note below)   1.00E-03       does another sub-outer
 *            eps   - fractional ptwise fission change        1.00E-03       does another sub-outer
 *                    per sub-outer (see note below)
 *
 *            note: eps, when the problem is finally converged, will equal epsi, the value shown above. however,
 *                  early in the iteration process, a larger value may be used to avoid unneccessary iterations.
 *
 *                                       final convergence criteria
 *                                       --------------------------
 *         criterion    quantity to test                          value     action taken if value exceeded
 *         ---------    ----------------                          -----     ------------------------------
 *            oitm  - outer iteration count                        11         quits with error message
 *            epsi  - transport lambda-1.0                    1.00E-03       does another outer
 *
 ********************************************************************************************************************************
 ********************************************************************************************************************************
1
 ********************************************************************************************************************************
 ********************************************************************************************************************************
 *
 *                     ...flux and eigenvalue convergence as monitored by partisn...
 *
 ********************************************************************************************************************************
 ******************************
 *key start iteration monitor *
 ******************************
 *
 *
 * cpu time outer       diffusion      k-eff                 max ptwise   max ptwise   inners
 *  (sec)    no. inners sub-outers  eigenvalue   lambda-1    flux change  fiss change converged
 *   0.40     0
 *   1.15     1      0       5      0.94625959  8.71687E-03  0.00000E+00  0.00000E+00  **no**
 *   3.11     2      4       2      0.97183687  2.04534E-02  2.41730E-01  2.60224E-01  **no**
 *   5.06     3      4       2      0.97317648  4.44334E-04  3.73616E-02  3.59170E-02  **no**
 *   7.00     4      4       2      0.97341081  5.81321E-05  6.55255E-03  6.50322E-03  **no**
 *
 *     ---------------------------------------------------------------------
 *
 *              -- inner iteration summary for outer iteration no.   5 --
 *
 *                                 iter per max flux   at
 *                              group  group   change   mesh
 *                                 1     1   0.12E-02   5, 12, 25
 *                                 2     1   0.71E-03   8, 11, 25
 *                                 3     1   0.44E-03   9,  3, 30
 *                                 4     1   0.25E-02   9,  3, 25
 *
 *
 * cpu time outer       diffusion      k-eff                 max ptwise   max ptwise   inners
 *  (sec)    no. inners sub-outers  eigenvalue   lambda-1    flux change  fiss change converged
 *   8.83     5      4       2      0.97346344  1.47826E-05  2.45206E-03  1.35800E-04  **no**
 *
 *  particle balance = -5.53556E-06     total inners all outers =   16
 *
 ********************************************************************************************************************************
1
 ********************************************************************************************************************************
 ********************************************************************************************************************************
 *
 *                                    ...group edit and balances upon convergence...
 *
 ********************************************************************************************************************************
 *
 *
 *                     ...title--- 3D model of a small FBR (Benchmark prob from Japan).                    ...
 *
 *
```

```
*                                        ...system balance tables...(neutrons only)
*
*
*****************************
*key start balance table     *
*****************************
*
*
*  gp          source    fission source        absorption        in scatter      self scatter       out scatter       net leakage
*
*   1   0.0000000E+00     5.8331900E-01     9.3605396E-02     4.4408921E-16     9.0151653E-01     4.7787507E-01     1.1841037E-02
*   2   0.0000000E+00     4.0545000E-01     2.4700661E-01     4.5277270E-01     1.5724706E+01     5.3335668E-01     7.7863085E-02
*   3   0.0000000E+00     1.1231000E-02     4.3337483E-01     5.5837083E-01     1.9303965E+01     6.5330568E-02     7.0894661E-02
*   4   0.0000000E+00     0.0000000E+00     6.0224027E-02     6.5357953E-02     9.8798048E-01     1.2076901E-06     5.1322510E-03
*
* tot   0.0000000E+00     1.0000000E+00     8.3421087E-01     1.0765015E+00     3.6918168E+01     1.0765635E+00     1.6573103E-01
*
*
*
*  gp   right leakage horizontl leakage       top leakage  vertical leakage      back leakage   fr-back leakage  particle balance
*
*   1   3.9867873E-03     3.9867873E-03     4.1668781E-03     4.1668781E-03     1.8436947E-03     3.6873716E-03    -4.2927836E-06
*   2   2.5616525E-02     2.5616525E-02     2.6517731E-02     2.6517731E-02     1.2864473E-02     2.5728829E-02    -7.1184843E-06
*   3   2.3916514E-02     2.3916514E-02     2.4463514E-02     2.4463514E-02     1.1257364E-02     2.2514632E-02    -4.9155671E-06
*   4   1.5828358E-03     1.5828358E-03     1.6027472E-03     1.6027472E-03     9.7333810E-04     1.9466680E-03    -1.2451109E-06
*
* tot   5.5102663E-02     5.5102663E-02     5.6750870E-02     5.6750870E-02     2.6938870E-02     5.3877500E-02    -5.5355618E-06
*
*
*
*  gp    left leakage    bottom leakage     front leakage     nprod spectrum     time source
*
*   1   0.0000000E+00     0.0000000E+00     1.8436769E-03     2.6283436E-01     0.0000000E+00
*   2   0.0000000E+00     0.0000000E+00     1.2864356E-02     3.9058318E-01     0.0000000E+00
*   3   0.0000000E+00    -3.4694470E-18     1.1257268E-02     3.0268793E-01     0.0000000E+00
*   4   0.0000000E+00     0.0000000E+00     9.7332993E-04     4.3894530E-02     0.0000000E+00
*
* tot   0.0000000E+00    -3.4694470E-18     2.6938630E-02     1.0000000E+00     0.0000000E+00
*
*****************************************************************************************************************************************
1
*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*    Multigrid work units...  Total=    584.08 WU.
*         By group...
*    1    121.86  2    160.51  3    160.51  4    141.19
*
*    Multigrid average convergence rate by group...
*    1 0.5118  2 0.5384  3 0.5134  4 0.5800


integral summary information
summary integral-k-eff     9.7346344E-01
summary integral-neutrons  1.5361768E+02
integral-source-i            neutron  0.0000000E+00
integral-time source-i       neutron  0.0000000E+00
integral-fission-i           neutron  1.0000000E+00
integral-absorption-i        neutron  8.3421087E-01
integral-in-scak-i           neutron  1.0765015E+00
integral-self-scak-i         neutron  3.6918168E+01
integral-out-scak-i          neutron  1.0765635E+00
integral-net lkage-i         neutron  1.6573103E-01
integral-right lkage-i       neutron  5.5102663E-02
integral-horizontal lkage-i  neutron  5.5102663E-02
integral-top lkage-i         neutron  5.6750870E-02
integral-vertical lkage-i    neutron  5.6750870E-02
integral-back lkage-i        neutron  2.6938870E-02
integral-fr-back lkage-i     neutron  5.3877500E-02
integral-particle bal-i      neutron -1.7571946E-05
*
*
*                                        ...interface file rtflux written..

*
*
*                                        ...interface file sncons written..

*
*****************************************************************************************************************************************

PARTISN solver iteration time       =         8.505 seconds
   mesh analysis and adjustments    =         0.003 seconds
   first collision source           =         0.000 seconds
   outer source setup               =         0.013 seconds
   inner source setup               =         0.011 seconds
   transport sweeps                 =         5.813 seconds
      angular source                =         0.038 seconds
      flux moments                  =         0.038 seconds
   transport synthetic acceleration =         0.000 seconds
   diffusion synthetic acceleration =         2.564 seconds
      diffusion source              =         0.053 seconds
      diffusion solve               =         1.423 seconds
         relaxation                 =         0.852 seconds
         interpolation              =         0.133 seconds
         condensation               =         0.172 seconds
         converge test              =         0.000 seconds
         diffusion setup            =         1.033 seconds
   particle balance                 =         0.007 seconds

Transport Grind Time                =       788.981 nanoseconds
Diffusion Grind Time                =       746.653 nanoseconds

*
*****************************************************************************************************************************************
1*****************************************************************************************************************************************
*****************************************************************************************************************************************
*
*
```

```
*                                        run highlights
*****************************
*key start run highlights    *
*****************************
*                                  ************************************
*                                  *                                  *
*                                  *    all modules are tentatively go.   *
*                                  *     interface file geodst written.   *
*                                  *       cross sections from cards.     *
*                                  *    *kwikrd overridden for ifido=2*   *
*                                  *    interface mixing files written.   *
*                                  *     interface file asgmat written    *
*                                  *    xs files macrxs,smxedt written.   *
*                                  *     interface file solimp written.   *
*                                  *   edit module execution suppressed.  *
*                                  *  neither editit nor edit cards exist.*
*                                  *        start solver execution.       *
*                                  *     interface file rtflux written.   *
*                                  *     interface file sncons written.   *
*                                  *                                  *
*                                  ************************************
*
*
*                                    ...execution terminated...
*
*
*******************************************************************************************************************************
*******************************************************************************************************************************
```

APPENDIX B: OPERATING SYSTEM SPECIFICS

UNIX/LINUX/UNICOS Execution

On UNIX, LINUX or UNICOS systems, the input is on STDIN and the printed output is on STDOUT. Thus, the user will normally cause execution of the program with the command:

partisn < *odninp* > *odnout*

where - partisn is the name of the executable file, *odninp* is the user's choice for a name for the input file, and *odnout* is the user's named output file. Whoever forms the executable names the executable file. The name customarily used is partisn.

STDERR contains a summary of the problem as it executes and, by default, is sent to the terminal screen. Also included on STDERR are any error messages.

Library Search Path

Most files read or written by PARTISN are in the current UNIX working directory. Some forms of cross-section files may be kept in other directories. By setting the environment variable SNXSPATH, the user may specify an ordered set of alternate directories in which the program should look for the named files. As an example, if an ISOTXS file is in the directory, **`/usr/tmp/xs`**, then the following command can be used

```
setenv SNXSPATH /usr/tmp/xs
```

and PARTISN will then look in that named directory for the library. The search path for each of the possible libraries is given in Table 2.2.

Table 2.2 UNIX Search Path

LIB	SEARCH PATH
MACRXS	Current Working Directory (CWD).
GRUPXS	**`SNXSPATH`**, then CWD.
ISOTXS	**`SNXSPATH`**, then CWD.
BXSLIB	**`SNXSPATH`**, then CWD, but see text below.
ODNINP	None, the library is contained in the input file.
MACBCD	CWD
XSLIBB	CWD
MENDF[a]	Path defined in the code on UNICOS. MENDF binaries are unavailable for other platforms.
MENDFG[a]	
XSLIB	**`SNXSPATH`**, then CWD
other	For any name other than those above, the program will assume the form is XSLIB and search for it in **`SNXSPATH`**, then CWD.

a. Available only at Los Alamos.

`SNXSPATH` can be used to protect an input BXSLIB file from being overwritten. See the discussion on page 2-46.

PARTISN CHAPTER INDEX

A

B

C

D

E

F

G

I

M

N

O

P

Q

R

S

T

U

V

X

Z